কম্পিউটার ও প্রোগ্রামিং



সৌম্য মিত্র

কম্পিউটার ও প্রোগ্রামিং

সৌম্য মিত্র

বি. বি. এস. পাবলিশার্স অ্যাণ্ড ডিস্ট্রিবিউটার্স
৮৮, মহাত্মা গান্ধী রোড
কলিকাতা-৭০০০০৯
ফোন : ৩১৩০৩১

Computer O Programming
by Soumya Mitra

প্রথম প্রকাশ
১লা বৈশাখ ১৩৯৪
এপ্রিল ১৯৮৭
দ্বিতীয় প্রকাশ
সেপ্টেম্বর ১৯৮৮
তৃতীয় প্রকাশ
সেপ্টেম্বর ১৯৯০

প্রকাশক :
রতন বিশ্বাস
বি. বি. এস. পাবলিশার্স অ্যাণ্ড ডিষ্ট্রিবিউটার্স
৮৮ এম. জি রোড
কলিকাতা-৭০০০০৯

প্রচ্ছদ, অলঙ্করণ ও ডায়াগ্রাম-অঙ্কন
সৌম্য মিত্র

ব্লক ও প্রচ্ছদমুদ্রণ
উজ্জ্বল দাশগুপ্ত
প্রসেস সিণ্ডিকেট
৫০ ভোলানাথ পাল লেন
কলিকাতা-৭০০০০৬

মুদ্রন :
দীপাশ্রী এন্টারপ্রাইজ
কলিকাতা-৭০০ ০০২

মূল্য : ৬৫.০০

কম্পিউটার ও প্রোগ্রামিং সম্বন্ধে বিশেষজ্ঞদের অভিমত

শিল্প বিপ্লবের ফলে অষ্টাদশ শতাব্দীতে যেমন উৎপাদনের ক্ষেত্রে এক যুগান্তকারী পরিবর্তন এসেছিল, কম্পিউটারের আবির্ভাবে আমরা সেই রকম আর এক বিপ্লবের সামনে এসে দাঁড়িয়েছি। শিল্পের নানা ক্ষেত্রে কম্পিউটারের প্রয়োগে কেবল যে উৎপাদনের পরিমাণই অনেক গুণ বেড়েছে তা নয়, উৎপন্ন সামগ্রীর গুণগত উৎকর্ষতাও বেড়েছে অনেক এবং শিল্পক্ষেত্রে অনেক নতুন সম্ভাবনার পথ খুলে গেছে।

এসব সত্ত্বেও আমাদের দেশে, বিশেষত দেশের শ্রমিক ও কর্মচারী মহলে, কম্পিউটার সম্পর্কে এখনও যথেষ্ট বিদ্বেষ ও ভীতি রয়েছে। এর প্রধান কারণ এই যে অনেক সময়ই কম্পিউটারকে এমন ক্ষেত্রে প্রয়োগ করা হয়েছে যেখানে সে মানুষের সঙ্গে প্রতিযোগিতা ক'রে তার কর্মহানি ঘটিয়েছে। কিন্তু মানুষকে যদি বোঝানো যায় যে কম্পিউটারের এমন অসংখ্য প্রয়োগক্ষেত্র আছে, যেখানে সে প্রতিযোগিতা না করে সাহায্যের হাত বাড়িয়ে দেয়, তখন মানুষ নিশ্চয়ই কম্পিউটারকে বন্ধু হিসাবে গ্রহণ করবে।

এর জন্য প্রথমেই প্রয়োজন কম্পিউটার সম্পর্কে ব্যাপক আলোচনা—কিভাবে কম্পিউটার কাজ করে, কি ভাবে তাকে দিয়ে কাজ করিয়ে নেওয়া হয়, ইত্যাদি। বর্তমানে আমাদের দেশে কম্পিউটার ও কম্পিউটার প্রোগ্রামিং সম্পর্কে নানা ধরনের স্বল্পকালীন প্রশিক্ষণের ব্যবস্থা হয়েছে, কিন্তু বাংলা ভাষায় এই বিষয়ের ওপর বই নেই বললেই চলে। ইংরাজী বইগুলিও অনেক ক্ষেত্রে আমাদের দেশের শিক্ষার্থী, বিশেষত যাঁরা স্কুল কলেজের গণ্ডীর বাইরে পাঠ নিতে চান, তাঁদের উপযোগী নয়। শ্রী সৌম্য মিত্র তাঁর রচিত গ্রন্থ মারফৎ এই অভাব দূর করেছেন। অত্যন্ত সাবলীল ভাষায় লেখা তাঁর এই বই পড়তে অসুবিধা হয় না, পরিভাষা নিয়ে তিনি মাথা ঘামাননি, কম্পিউটার জগতে যেসব ইংরাজী শব্দ বহুল পরিমাণে ব্যবহৃত হয়, তিনি সরাসরি সেই সব শব্দ ব্যবহার করেছেন। তাঁর এই বই কম্পিউটার সম্পর্কে অনেকেরই কৌতূহল চরিতার্থ করবে এবং কম্পিউটার ব্যবহারে মানুষকে উৎসাহিত করবে, একথা নিঃসন্দেহে বলা যায়।

শ্রী মিত্রের এই সাধু প্রচেষ্টার সর্বাঙ্গীন সাফল্য কামনা করি।

অনাদি নাথ দাঁ
প্রাক্তন প্রধান অধ্যাপক, রেডিও ফিজিক্স ও
ইলেকট্রনিক্স বিভাগ, কলিকাতা বিশ্ববিদ্যালয়

The book titled "Computer" has been nicely written by Soumya Mitra. A Complex subject like computer has been presented in an easily understandable manner. I am sure it would go a long way to benefit its readers.

Ratneswar Ray
30. 3. 87.
Sr. Engineer, Saha Institute of Nuclear Physics.
Guest Lecturer, Dept. of App. Physics, Calcutta University

বাংলা ভাষায় প্রযুক্তিবিদ্যার বইয়ের খুবই অভাব। বিশেষ করে কম্পিউটার বা ডিজিটাল ইলেক্ট্রনিক্সের বিষয়ের ওপর বই খুঁজেই পাওয়া যায় না। এই হতাশ করা পরিবেশের মধ্যে আশার আলো জ্বালাতে পারবে এই "কম্পিউটার ও প্রোগ্রামিং" নামক বইটি।

এই বইটা পড়ে উপকৃত হবেন, একজন বিজ্ঞ থেকে একজন অজ্ঞ পর্যন্ত সকলেই। বিশেষ করে যাঁরা কম্পিউটারকে ব্যবসায়িক দৃষ্টিভঙ্গি ছাড়াও অন্যভাবে দেখতে চান, তাঁদের পক্ষেও উপযুক্ত হবে এই বইটি। এই বইটিতে Hardware ও Software এই দুই বিষয়ের ওপর অনেক কিছু তথ্য দেওয়া হয়েছে যার ফলে ভারত সরকারের Class Project-এর জন্যে স্কুলের Computer-literacy programme-এর পাঠ্য হিসেবে এই বইকে সুপারিশ করা উচিত। এছাড়া এতে BASIC-Language-এর উদাহরণ স্বরূপ B. B. C. Micro computer-এ লেখা ভাল ভাল Listing যোগ করা হয়েছে যা অনেক বিদেশী বইতে পাওয়া যাবে কিনা সন্দেহ আছে।

C. S. Roy
Scientific Adviser (Full-timer)
Hindu School Science Centre

প্রসঙ্গকথা

পাঠক সাধারণই গ্রন্থ রচনার প্রেরণারূপে গ্রন্থকারকে কোথাও কোনো বই লেখায় উৎসাহিত করেছেন কিনা জানিনা, তবে আমার পরম সৌভাগ্য যে বর্তমান বইটি রচনার মূল প্রেরণা আমি আমার অসংখ্য প্রিয় পাঠক পাঠিকার কাছেই পেয়েছি। এই সুযোগ করে দিয়েছে আমারই সম্পাদিত ছবি ইলেকট্রনিক্স ও ফোটোগ্রাফির মাসিক পত্র 'ফোটন'। এই পত্রিকার সম্পাদনা সূত্রে ফোটন পাঠকদের কাছ থেকে যে অজস্র চিঠি আমি রোজ দিন পেয়ে থাকি এবং যে সব চিঠি আসে ভারতের বিভিন্ন রাজ্য ও প্রতিবেশী রাজ্য বাঙলাদেশ, নেপালের বঙ্গভাষাভাষী ফোটন বন্ধুদের কাছ থেকে, তারই মধ্যে আমি কম্পিউটার সম্পর্কে তাঁদের আগ্রহের কথা জেনেছি। ইলেকট্রনিক্সের নানা বিষয়ের সঙ্গে তাঁরা কম্পিউটার এবং তার প্রোগ্রামিং সম্পর্কে সহজ বাঙলায় পূর্ণাঙ্গ আলোচনার জন্য অনুরোধ জানাতে থাকেন। 'ফোটন' পত্রিকায় নিজেই নিজের কম্পিউটার কিভাবে বানানো যেতে পারে তার সার্কিট সহ আলোচনা তখনই শুরু করে দিই। এই ধারাবাহিক আলোচনায় যোগান দেন হিন্দু স্কুল সায়ান্স সেন্টারের পরিচালক ও শিক্ষক চন্দ্রশেখর রায় বা আমাদের শ্রদ্ধেয় চাঁদুদা।

বলতেও ভালো লাগে যে বঙ্গভাষী ইলেকট্রনিক্স অনুরাগী সমাজ আজ সাগ্রহে সকল বিষয়ে জানতে ও বুঝতে চাইছেন। তাই তাঁদের কাছ থেকে কম্পিউটার বিষয়ে একটি পূর্ণাঙ্গ গ্রন্থ রচনা করে দেওয়ার অনুরোধ আসতে থাকে। তাঁদের অনেকের বক্তব্য, ইংরেজিতে কম্পিউটার সফট্ওয়্যার এবং হার্ডওয়্যারের ওপর স্বতন্ত্রভাবে অনেক ভালো বই আছে, নেই কোথাও একত্রে দুটি বিষয়ের সহজবোধ্য পূর্ণাঙ্গ আলোচনা। তাছাড়া যাঁরা শিক্ষার্থী, মাতৃভাষার মাধ্যমে শিখতে পারেন এমন একটি পূর্ণাঙ্গ পাঠ্যপুস্তকের অভাব তাঁরা বিশেষভাবে অনুভব করছেন এবং সেই অভাবটি তাঁদের পূরণ করে দিতে হবে। পত্র লেখকদের মধ্যে স্কুল কলেজের এবং কম্পিউটার প্রশিক্ষণ কেন্দ্রের ছাত্রছাত্রী যেমন আছেন, তেমনিই আছেন তরুণ তরুণী যুবক যুবতী এমন কি বৃদ্ধ বয়স্ক শিক্ষার্থী ও জিজ্ঞাসু পাঠক।

কাজটি মোটেই সহজ নয়, তাছাড়া এমন একটি পূর্ণাঙ্গ গ্রন্থের কলেবর যে কতদূর স্ফীতিলাভ করবে তার আন্দাজ শুরুতে করে ফেলাও ছিল অসম্ভব। সুতরাং এমন কাজে হাত দেওয়ার চেষ্টা অকল্পনীয়ই থেকে যেত যদি না বিশ্বাস বুক স্টল-এর রতন দা আমাকে সবুজ সংকেত দিয়ে বলতেন, যেমনটি দরকার লিখে যান, আপনার পাঠকদের হাতে বই তুলে দেওয়ার বাকি দায়িত্ব আমার। বই কেমন হয়েছে তা পাঠকেরই বিচার্য, তবে যা হয়েছে তা বাজারে হাজির করার যে অসম সাহসিকতা শ্রীরতন বিশ্বাস দেখিয়েছেন তা বস্তুতই প্রশংসনীয়। তাঁকে এই সুযোগে জানাই আমার আন্তরিক কৃতজ্ঞতা, শুধু এই কারণেই যে, তিনি আলোচ্য বিষয়ে পাঠকদের অভাব পূরণের প্রশ্নটি তাঁর লাভালাভের প্রশ্নের থেকে বড় করে দেখেছেন।

চন্দ্রশেখর রায় বা চাঁদুদার উদয়াস্ত সাধনা ইলেকট্রনিক্স অনুরাগী ছাত্র বন্ধুদের নিয়ে। এই বই লিখছি শুনে তিনিও উৎসাহিত করেছেন। বইটি লেখাকালীন আমার নিজের কম্পিউটারটি হঠাৎ

বিকল হয়ে পড়লে, তাঁর সাম্রাজ্যে বসে কম্পিউটার ব্যবহারের সুযোগ দিয়ে তিনি আমাকে কৃতজ্ঞ করেছেন অনেক।

এই সব কিছুর মূলে, আগেই বলেছি, বিভিন্ন শ্রেণীর পাঠক পাঠিকার তাগিদ কাজ করেছে। তাই আমাকে এমন ভাবে বইটির বিষয়াবলী উপস্থিত করতে হয়েছে যার দ্বারা সব বয়সের এবং সব মানের শিক্ষার্থীই বইটি থেকে তাঁর প্রয়োজনীয় আলোচ্য বিষয় পেয়ে যান এবং যাতে কোথাও কোনো রকম দুর্বোধ্যতার অনুপ্রবেশ না ঘটে। বিশেষতঃ শিক্ষার্থী যাতে নিজে নিজেই বিষয়টি আয়ত্ত করতে পারেন সবিশেষ লক্ষ্য সেদিকেও রাখতে হয়েছে। কেননা আমার প্রেরণাদাতাদের মধ্যে এমন অনেক বয়স্ক পাঠক আছেন, আছেন এমন আরও পাঠক পাঠিকা যাঁদের পক্ষে কোনোও প্রশিক্ষণ কেন্দ্রে যাওয়ার সুযোগ নেই অথচ জানার আগ্রহ আছে তাঁদের ষোলো আনা।

লক্ষ্য রাখতে হয়েছে সব চেয়ে জরুরী আর একটি বিষয়ের প্রতি। বিজ্ঞ শিক্ষার্থীরা নিশ্চয় জানেন, ইংরেজি ভাষায় কম্পিউটার প্রযুক্তিবিদ্যা বা হার্ডওয়্যার এবং তার ব্যবহারিক দিক সফট্ওয়্যার সম্পর্কে স্বতন্ত্র ভালো বই থাকলেও একই আধারে দুটি বিষয়ের পূর্ণাঙ্গ আলোচনা সহজলভ্য নয়। ভালো বই আছে কিন্তু বেশীর ভাগই সেগুলি শ্রেণী বিভক্ত, অর্থাৎ হয় কোনোটি একেবারেই প্রাথমিক ও প্রবেশিকা পর্যায়ের আলোচনায় সীমাবদ্ধ, নয়তো জ্ঞানী প্রযুক্তিবিদদের উপযোগী করে লেখা। এমন শ্রেণীবিভক্ত পাঠ্য বই রচনার সুযোগ কিন্তু আমার পাঠকরাই আমাকে দেন নি। আগেই জানিয়েছি তাঁরা পূর্ণাঙ্গ পাঠ্য বইয়ের দাবী করেছেন এবং আমি তা যথাসাধ্য পূরণ করার চেষ্টা করেছি। সফল কতটা হয়েছি তা অবশ্য গুণীজনেই বিচার করবেন। সুতরাং কম্পিউটার ও প্রোগ্রামিং সাক্ষরতার ক্ষেত্রে এ বইটির এইখানেই বিশেষ স্বাতন্ত্র্য রক্ষার চেষ্টা করেছি। পাঠক যাতে কম্পিউটারের যান্ত্রিক প্রযুক্তি সম্বন্ধে একটি স্বচ্ছ সুস্পষ্ট ধারণা গড়ে নিতে পারেন এবং একই সঙ্গে শিখতে পারেন BASIC ভাষায় প্রোগ্রাম তৈরীর পদ্ধতি সেজন্য সবিশেষ প্রযত্নের ত্রুটি রাখি নি।

ভাষার ক্ষেত্রে প্রধানত আন্তর্জাতিক পরিভাষা ব্যবহার করে প্রয়োজনকে স্বীকৃতি জানিয়েছি। কম্পিউটার সাক্ষরতার প্রয়োজনে যে ভাষা জানা ও প্রয়োগ করা অনিবার্য তাকে ভাষান্তরিত করে পাঠের সহজ প্রবাহপথ রুদ্ধ করার পক্ষে কোনো যুক্তি নেই। যদিও পরিভাষার প্রয়োজনটিও একেবারেই অগ্রাহ্য করার বিষয় নয়। আর সেজন্যই গ্রন্থের শেষে একটি ইংরেজি বাংলা কম্পিউটার অভিধান সংযোজন করে সেই অভাব পূর্ণ করার চেষ্টা করেছি।

যদি বইটিকে কম্পিউটার সাক্ষতার ক্ষেত্রে পাঠ্য পুস্তকরূপে দেখতে চাই, আমার মনে হয়েছে, হার্ডওয়্যার বা প্রাযুক্তিক বিষয়টির নানান জটিল তত্ত্ব তাহলে এই গ্রন্থে পরিহার করা দরকার। সর্বাধুনিক ইলেকট্রনিক্স প্রযুক্তির সঙ্গে যাঁরা অপরিচিত তাঁদের জন্য বিষয়টির সহজবোধ্য উপস্থাপনাই অধিকতর প্রয়োজন। এটা আমার নিজস্ব অভিজ্ঞতার ফলশ্রুতি। হার্ডওয়্যার নিয়ে কাজ করার সময় বুঝেছি, প্রাথমিক শিক্ষায় প্রাযুক্তিক জটিলতা বিষয়প্রবেশে অনীহা ও দুর্বোধ্যতারই সৃষ্টি করবে। সুতরাং অকারণ জটিলতা পরিহার করেছি।

সব ক্ষেত্রেই সচিত্র উদাহরণ ব্যবহার করেছি যাতে 'নিজে শেখা'র সুযোগ সমান বর্তমান থাকে।

পরিশেষে আর একটি কথা জানিয়ে বর্তমান প্রসঙ্গ শেষ করব। এই বইয়ে BASIC ভাষায় এ্যাপ্লিকেশন সফট্‌ওয়্যারের আলোচনার ওপরই জোর দেওয়া হয়েছে। Enhanced BASIC-এর Keyword-গুলির আলোচনা এখানে বাহুল্যবোধে পরিত্যক্ত হয়েছে। মূল BASIC [বা Standard BASIC], যা সব কম্পিউটারেরই সাধারণ ভাষা তাই আলোচিত হয়েছে। Enhanced BASIC ভাষা বিভিন্ন কোম্পানীর নিজস্ব কম্পিউটারের ভাষা যা শিক্ষার্থীর শিক্ষার পক্ষে অপরিহার্য নয়। যাঁরা বিশেষ কোনো Enhanced BASIC Keyword জানতে আগ্রহী তাঁরা সংশ্লিষ্ট কোম্পানীর ম্যানুয়ালেই সেই Keyword-এর Syntax-টি শিখে নিতে পারবেন। কোনো বিশেষ সংস্থার Enhanced BASIC ব্যবহার তাই অসমীচীন বিবেচনায় পরিহার করেছি।

আমার সকল পরিশ্রম উজাড় করে ঢেলে দিয়েও, নিশ্চয় জানি, সকলের সব অভাব পূরণ করতে পারি নি। একটি খণ্ডে তা সম্ভবও নয়। তবে সুযোগ হলে এবং পাঠক সাধারণের উৎসাহ পেলে পরবর্তী পর্যায়ে অপর একটি খণ্ড রচনা করে ফাঁক ভরাবার চেষ্টা করব।

১৫/২ রাণীশঙ্করী লেন, কলি-২৬

সৌম্য মিত্র

সূচীপত্র

পরিচ্ছেদ ১

কম্পিউটারের ক্রমবিকাশ

কম্পিউটার শব্দটির সঙ্গে অনেক রহস্যই জড়িয়ে রয়েছে। সায়ান্স ফিকশন কমিকস্‌, সিনেমায় আর গল্পে যে সব কাল্পনিক কম্পিউটারের সঙ্গে আমাদের পরিচয় ঘটে তারা প্রত্যেকেই প্রায় যন্ত্রদানব জাতীয় উদ্ভট জিনিস। মানুষের মতো গলার স্বরে কথা বলতে পারে, যখন কাজ করে তখন হাজার হাজার রঙিন বাতি জ্বলতে নিভতে থাকে। প্রয়োজনে লেসার অস্ত্র চালায়, মাঝে মাঝে রেগে গেলে বিদ্যুতের ঝলকানি আর ধোঁয়া বের হয়। সব মিলিয়ে সে এক দারুণ ব্যাপার। এই সমস্ত চমকপ্রদ ঘটনা সিনেমায় দেখে, গল্পে পড়ে আর কানে শুনে কার না মনে কম্পিউটার সম্বন্ধে ভয় মেশানো শ্রদ্ধা সৃষ্টি হয়? কাগজে এবং পত্র-পত্রিকায় বিভিন্ন কম্পানির কম্পিউটারের বিজ্ঞাপন পড়ে মনে হয় কম্পিউটার হলো এক অত্যাশ্চর্য জটিল ইলেকট্রনিক মস্তিষ্ক যার টেকনিক্যাল ব্যাপার বোঝা সাধারণ মানুষের কর্ম নয়।

অথচ যাঁরা কম্পিউটার নিয়ে দৈনিক কাজকর্ম করেন, তাঁরা ভাল করেই জানেন কম্পিউটার সম্বন্ধে প্রচলিত এ সমস্ত ধারণাই একদম মিথ্যে। কম্পিউটার মোটেই দুর্বোধ্য যন্ত্র নয়। যাঁরা বিজ্ঞান কিছুই জানেন না, তাঁরাও অনায়াসে বাহ্যিক ভাবে একটা কম্পিউটার কি করে কাজ করে তা বুঝতে পারেন। আর যাঁরা বিজ্ঞানের ছাত্র, তাঁদের পক্ষে কম্পিউটারের কাজ বোঝা তো আদৌ কোন কঠিন ব্যাপারই নয়।



কম্পিউটারের প্রথম অবস্থা

কম্পিউটার একটি বুদ্ধিমান ইলেকট্রনিক মস্তিষ্ক যেটি নিজে নিজে চিন্তা ভাবনা করে বিরাট বিরাট সমস্যা নিমেষে সমাধান করে দেয়, এ ধরনের প্রচলিত ধারণা যে আলাদিনের আশ্চর্য প্রদীপের মতই গাল-গল্প একথা কম্পিউটার প্রোগ্রামাররা প্রতিদিন হাড়ে হাড়ে উপলব্ধি করেন। তথাকথিত এই বুদ্ধিমান যন্ত্রকে দিয়ে একটা বড় সড় কাজ করিয়ে নিতে সুশিক্ষিত প্রোগ্রামার আর বিজ্ঞানীরা এয়ার কণ্ডিশন করা ঘরে বসেও যে প্রায়ই ঘেমে নেয়ে চান করে যান, সে খবর আর কজন রাখেন ?

মজার ব্যাপার হলো, যে যন্ত্রটিকে ঘিরে এতো গাল-গল্প আর জল্পনা কল্পনা গড়ে উঠেছে সেটি কিন্তু খুব একটা আধুনিক আবিষ্কার নয়। কেটলি থেকে বেরোনো স্টিম, যাকে দেখে বিজ্ঞানী জেমস্ ওয়াটের মনে স্টিম ইঞ্জিন ডিজাইন করার চিন্তা জাগে তাকে যদি স্টিম ইঞ্জিনের পূর্বসূরী ধরা যায়, তবে সেই হিসেবে কম্পিউটারের পূর্বসূরীর বয়স কম করেও সাড়ে তিন হাজার বছর হবে। আর যদি জেমস ওয়াটের 'রকেট'কে প্রথম স্টিম ইঞ্জিন ধরা যায়, তবে প্রথম কম্পিউটারের মডেলটি তৈরী হয়েছিল একশো বছর আগে।

কম্পিউটারের মতোই কম্পিউটর তৈরী হওয়ার ইতিহাসও যথেষ্ট আকর্ষণীয়। কম্পিউটারের ইতিহাস থেকেই জানা যায়, শুঁয়োপোকা থেকে প্রজাপতির বিবর্তনের মতোই বর্তমানের আধুনিক মাইক্রোপ্রসেসর দিয়ে তৈরী কম্পিউটারও নানা বিচিত্র বিবর্তনের মধ্য দিয়েই বর্তমান রূপ গ্রহণ করেছে।

চার হাজার বছর আগের গণনা যন্ত্র

চীন দেশে আজ থেকে হাজার চারেক বছর আগে একটা গণনা কার্যে সহায়ক যন্ত্র তৈরী হয়েছিল। নাম ছিলো তার অ্যাবাকাস (Abacus)। ছিলো বললে ভুল হবে, এই বিংশ শতাব্দীতে যখন আমরা রিস্ট ওয়াচ ক্যালকুলেটর নিয়ে কাজ করছি, তখনো অ্যাবাকাসের ব্যবহার অবিচ্ছিন্ন ভাবে চালু রয়েছে পৃথিবীর বিভিন্ন কোণে, বিভিন্ন দেশের বিভিন্ন বয়সের মানুষের মধ্যে।

অ্যাবাকাস আমরা সবাই দেখেছি। বাচ্চাদের প্যারাম্বুলেটরে স্লেটের সঙ্গে অথবা খেলনা হিসেবে আজও অ্যাবাকাস কিনতে

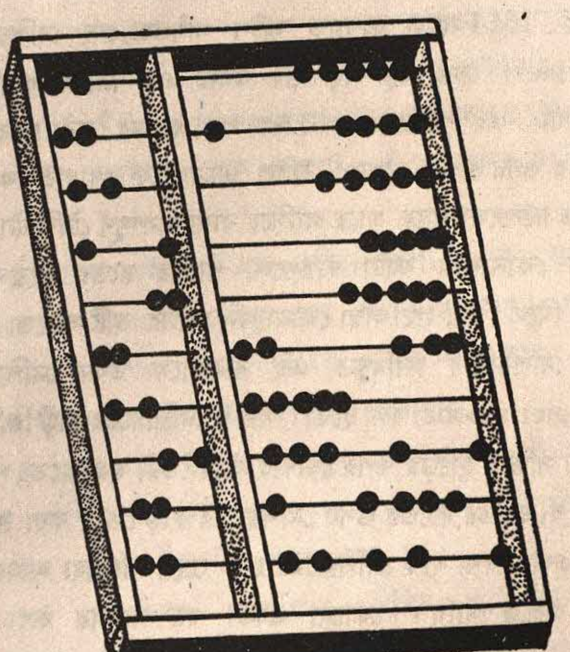
পাওয়া যায়। একটা কাঠের বা প্লাস্টিকের ফ্রেমে অনেকগুলো সমান্তরাল রড লাগানো থাকে। সেই রডে গলানো থাকে গোল গোল রঙিন কাঠের বল। ফ্রেমটার দুটো ভাগ থাকে। বড় ভাগটার প্রতিটি রডে পাঁচটা করে আর ছোট ভাগটার প্রতিটি রডে দুটো করে বল গলানো থাকে। এই খেলনাটার সঙ্গে আমরা সবাই পরিচিত। এখন ছোটরা এই নিয়ে খেলা করে, কিন্তু আসলে এটাই হোলো সেই চার হাজার বছর আগেকার অ্যাবাকাস।

ফ্রেমের বড় অংশটির রডে গলানো প্রতিটি বল 'এক' সংখ্যাকে বোঝায়, আর ছোট অংশের রডে গলানো প্রতিটি বল 'পাঁচ' সংখ্যাকে বোঝায়। সাধারণ অবস্থায় সব বলগুলিই ফ্রেমের কিনারার দিকে সরানো থাকে। অঙ্ক কষার সময় ওই জায়গা থেকে বলগুলিকে প্রয়োজন মতো ফ্রেমের বিভাজক পার্টিশনটির দিকে সরিয়ে আনা হয়। যেমন বড় অংশের রড থেকে দুটো বল ওই বিভাজক অংশের দিকে সরিয়ে আনলে $1+1=2$ বোঝায়। ঠিক তেমনি ছোট অংশ থেকে



অ্যাবাকাস

চিত্র ১



অ্যাবাকাস

অ্যাবাকাস কথাটির উৎপত্তি গ্রীক শব্দ অ্যাবাক্স (ABAX) থেকে। গ্রীক ভাষায় অ্যাবাক্স বলতে বোঝায় একটি সমতল বোর্ড।

প্রাচীন কালে অ্যাবাকাসের চেহারাটি ছিলো অন্যরকম। সে সময়ে সমতল একটি ট্রেতে বালি ঢেলে তার ওপর কাঠি দিয়ে আঁচড় কেটে গণনা করা হতো, আর এই ট্রেরই নাম ছিলো অ্যাবাকাস।



নেপিয়ারের হাড়

একটা বল বিভাজকের দিকে সরালে 5 বোঝায়। এইভাবে রডে গলানো বলগুলো সরিয়ে সরিয়ে এতে খুব তাড়াতাড়ি যোগ করা যায়।

1946 সালে একটা মজার প্রতিযোগিতা হয়েছিল। মাৎসুজা নামে এক জাপানী ভদ্রলোকের বিরুদ্ধে প্রতিযোগী ছিলেন এক অ্যামেরিকান ভদ্রলোক। মাৎসুজার হাতে ছিলো একটা অ্যাবাকাস. আর অ্যামেরিকান ভদ্রলোকের হাতে একটা ইলেকট্রনিক ক্যালকুলেটর। দুজনকেই একটা বিরাট যোগ করতে দেওয়া হলো। দেখা গেলো মাৎসুজা অ্যাবাকাসের সাহায্যে ক্যালকুলেটরের থেকেও দ্রুত যোগফল বার করে ফেললেন!

এই অ্যাবাকাসকেই কম্পিউটার এবং ক্যালকুলেটরের প্রথম পূর্বপুরুষ বলা চলে।

জন নেপিয়ারের হাড়

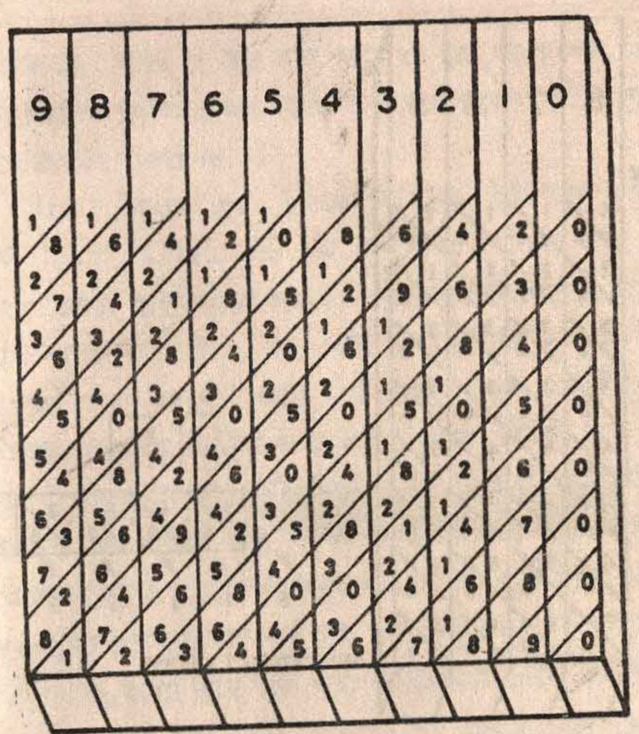
অ্যাবাকাসের আবিষ্কারে বড় বড় যোগ করা সহজ হয়ে গিয়েছিলে। কিন্তু দ্রুত গুণ করার দরকার পড়লে অ্যাবাকাস বিশেষ কাজে লাগতে না। 1614 সালে প্রখ্যাত স্কটিশ গণিতজ্ঞ জন নেপিয়ার (John Napier) দ্রুত বড় বড় গুণ করার এক বিশেষ পদ্ধতি আবিষ্কার করলেন। এই পদ্ধতিতে দশটা লম্বা লম্বা হাড়ের তৈরী আয়তক্ষেত্রাকার রডের ওপর একটা সংখ্যার টেবিল এমনভাবে খোদাই করা থাকতো যাতে রডগুলো গায়ে গায়ে লাগিয়ে রাখলে সম্পূর্ণ টেবিলটা পড়া যেতো আর কোনাকুনি কাটা দাগগুলো পাশের রডের গায়ে কাটা দাগের সঙ্গে মিলে গিয়ে লম্বা লম্বা কোনাকুনি দাগের সৃষ্টি করতো।

নেপিয়ারের আবিষ্কৃত এই ব্যবস্থাকে তখন নেপিয়ারের হাড় Napier's Bone) বলা হতো। যদিও নেপিয়ারের হাড় জিনিসটি প্রথমে সত্যি সত্যিই হাড়ের ওপর খোদাই করে তৈরী করা হতো, পরবর্তীকালে কাঠ বা হাতির দাঁতের ওপর নেপিয়ারের হাড় তৈরী করা হয়েছিল।

এবার দেখা যাক নেপিয়ারের হাড় যন্ত্রের সাহায্যে আমরা কি করে গুণ করতে পারি। কোনো একটা বড় সংখ্যার কথা চিন্তা করা যাক। ধরা যাক আমাদের 4765 এর সঙ্গে 7কে গুণ করতে হবে। আমরা তাহলে নেপিয়ারের হাড় থেকে 8, 7, 6 আর 5 সংখ্যার হাড়



নেপিয়ারের হাড়



চিত্র : ২ নেপিয়ারের হাড়

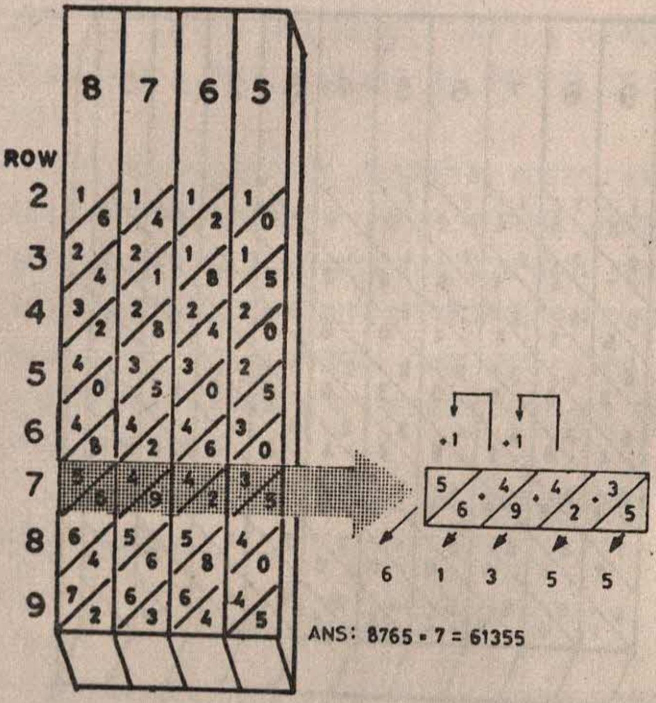
চারটে বেছে নিয়ে 8, 7, 6, 5 হিসাবে পাশাপাশি সাজাবো। তারপর 7 নম্বরের সারিতে (Row) যে সব সংখ্যা আছে সেগুলোকে ওই কোনাকুনি কাটা লাইন অনুসারে যোগ করলেই সঠিক গুণফল 61355 পেয়ে যাবো।

নেপিয়ারের এই পদ্ধতিতে সুবিধা হলো এখানে দুটো সংখ্যার গুণফল বার করতে গেলে কেবল যোগ করতে হয় আর যোগ করার সময় Cairy over হিসেবে হাতে একের বেশী সংখ্যা থাকে না। তাই যোগটাও মুখে মুখে খুব তাড়াতাড়ি করা যায়।

কিন্তু নেপিয়ারের হাড়ের সাহায্যে কেবলমাত্র গুণই করা যায়। অন্যান্য গাণিতিক ক্রিয়া, যেমন ভাগ, বর্গফল (Squre), বর্গমূল (Square root) ইত্যাদি করা যায় না। 1617 সালে জন নেপিয়ার নেপেরিয়ন লগারিদম গণিত আবিষ্কার করলেন। লগারিদমের (Logarithm) সাহায্যে খুব অল্প সময়ে বড় বড় গুণ, ভাগ, n^{th} power, n^{th} root বার করে ফেলা যায়। এর জন্য



স্লাইড রুল



চিত্র ৩

নেপিয়ার হাড়ের সহজে সাহায্যে গুণ করার পদ্ধতি

দরকার হয় একটা লগ এবং অ্যান্টিলগ টেবিল। পকেট ক্যালকুলেটর তৈরী হবার আগে পর্যন্ত সমস্ত ইঞ্জিনিয়ারিং-এর অঙ্ক সমাধান করতে লগ টেবিলেরই সাহায্য নেওয়া হতো। এখনও লগ টেবিলের সাহায্যে স্কুল আর কলেজের ছাত্ররা বড় বড় গুণ, ভাগ করে। এছাড়া গণিতের বিভিন্ন পর্যায়ে এবং বিজ্ঞানের বিভিন্ন শাখায় লগারিদমের ব্যবহার প্রচুর।

স্লাইড রুল

নেপিয়ারের লগারিদমের সাহায্যে গুণ, ভাগ, বর্গমূল ইত্যাদি বার করতে হলে উত্তরটা তৎক্ষণাৎ পাওয়া যায় না। এর জন্য প্রথমে লগ টেবিল থেকে কষতে দেওয়া অঙ্কের লগ বার করতে হয়। তারপর অঙ্ক অনুযায়ী যোগ বা বিয়োগ করে সেই যোগফল বা বিয়োগ ফলটি যে সংখ্যার লগ সেই সংখ্যাটি অ্যান্টিলগ টেবিল থেকে বার করতে হয়। তারপর দশমিক চিহ্নটি কোন জায়গায় বসবে সেটাও হিসেব করে বার

করতে হয়। যদিও এ সমস্ত কাজ করতে যে সময় লাগে সেটা, অঙ্কটিকে যদি সাধারণ পদ্ধতিতে করা হতো, তার থেকে অনেক কম, তবুও সঙ্গে সঙ্গে উত্তরটা পাওয়া যায় না।
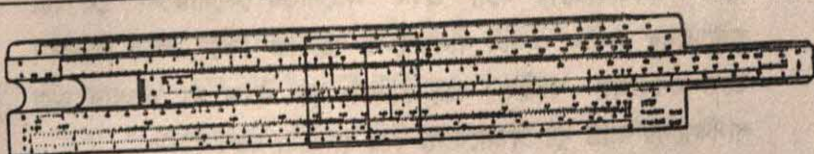


স্লাইড রুল

1620 সালে এক ইংরেজ গণিতজ্ঞ উইলিয়াম অগট্রেড (William Oughtred) ওই লগারিদম পদ্ধতির ওপর নির্ভর করে একটা স্লাইডিং স্কেল বানান। স্কেলটার দুটো অংশ। একটি সরু এবং অপরাংশ মোটা। বড় একটা চওড়া স্কেলের মাঝে একটি কাটা খাজের মধ্যে বসানো থাকে সরু স্কেল। সরু স্কেলটাকে ওই খাঁজের মধ্যে বড় স্কেলটার ওপর এপাশ ওপাশ স্লাইড করে সরানো যায়। বড় স্কেলে এবং ছোট স্কেলে যোগ বিয়োগ গুণ ভাগ, বর্গমূল ইত্যাদি বার করার জন্য বিভিন্ন সংখ্যা খোদাই করা থাকে। এই স্কেলের সাহায্যে আধুনিক ক্যালকুলেটরের মতই যে কোন যোগ, বিয়োগ, গুণ বা ভাগফল সঙ্গে সঙ্গে বার করা যায় যদিও আধুনিক ক্যালকুলেটরে যত বড় বড় সংখ্যা নিয়ে কাজ করা যায়, স্লাইড রুলে তত বড় সংখ্যা দিয়ে অঙ্ক কষা যায় না। তাছাড়া এই স্লাইড রুলে যে উত্তরটা পাওয়া যায় তাকে স্কেলের গায়ে খোদাই করা টেবিল থেকে পড়তে হয়। পড়বার সময় কিছু ভুল ভ্রান্তি থেকে যেতে পারে যেটা আধুনিক ক্যালকুলেটরের ডিজিটাল ডিসপ্লেতে হওয়ার কোনো সম্ভাবনা নেই। তবু চট্‌ করে হিসেব করার জন্য স্লাইড রুল অত্যন্ত জনপ্রিয় যন্ত্র হয়েছিল।

পকেট ক্যালকুলেটর যখন এতো সস্তা হয়নি, তখনো সমস্ত কলেজ এবং বিশ্ববিদ্যালয়ের ছাত্ররা হিসেব করার কাজে স্লাইড রুলই ব্যবহার করতেন। স্লাইড রুল নামটাও এতো জনপ্রিয় ছিলো যে কিছু কিছু আধুনিক ইলেকট্রনিক ক্যালকুলেটরের নির্মাতারা তাঁদের ক্যালকুলেটরের নাম ইলেট্রনিক স্লাইড রুল রেখেছিলেন।

চিত্র : ৪



স্লাইড রুল



পাসকালের যন্ত্র গণক

স্লাইড রুল সুবিধাজনক হলেও এটাকে ব্যবহার করতে রীতিমত অভ্যাস থাকা চাই। সুতরাং আরো সহজ যন্ত্র বানানোর দিকে সবাই মনোযোগ দিলেন। স্লাইড রুল তৈরীর বাইশ বছর পরে 1942 সালে বিখ্যাত ফরাসী গণিতজ্ঞ এবং দার্শনিক পাসকাল (Blaise Pascal) মাত্র উনিশ বছর বয়সেই এমনি একটা যন্ত্রগণক তৈরী করে ফেললেন। পাসকাল যন্ত্রটা তৈরী করেছিলেন তাঁর বাবার জন্য। পাসকালের বাবা সরকারী কর দপ্তরে কাজ করতেন আর তাঁকে সব সময়ই বিরাট বিরাট যোগ আর বিয়োগ করতে হতো। কাজের চাপও ছিলো দারুণ, তাই বাবার কাজের চাপ কমাতেই পাসকালের এই আবিষ্কার।

পাসকালের যন্ত্রগণকটি ছিলো পূর্ববর্তী সমস্ত যন্ত্রগণকের থেকে আলাদা। অ্যাবাকাসের মতো এতে কোনো বল ছিলো না, নেপিয়ারের যন্ত্রের মতো কোনো চ্যাপ্টা আয়তক্ষেত্রাকার হাড় ব্যবহার করা হয়নি, এমনকি স্লাইড রুলের মতো এতে কোনো স্কেলও ছিলো না। যন্ত্র গণকের মধ্যে ছিলো কেবল নানা আকৃতির পিন লাগানো চাকতি যেগুলো গিয়ারের (Gear) মতো কাজ করতো। যন্ত্রটার ওপরে ছিলো অনেকগুলো ডায়াল আর ছিলো ছোটো ছোটো জানলা। জানলাগুলোর ঠিক পিছনে ছিলো 0 থেকে 9 লেখা চাকতি (যেমন ট্যাক্সির মিটারে থাকে)। যখন কোনো অঙ্ক কষতে হতো তখন ওই ডায়াল ঘুরিয়ে যন্ত্রের জানলায় অঙ্কের সংখ্যাগুলো আনতে হতো। গিয়ারের সাহায্যে এমন ব্যবস্থা ছিলো যাতে এককের জানলার পিছনে যে সংখ্যা লেখা চাকতিটা লাগানো সেটা এক পাক ঘুরলেই দশকের জানলার পিছনের চাকতিটা এক ঘর সরে যেতো (যেমন ট্যাক্সির মিটারে হয়)।

কার্যগত দিক থেকে পাসকালের এই যন্ত্র গণক মোটেই জটিল ছিলো না। তখনকার দিনে এটা দারুণ জনপ্রিয়তাও লাভ করেছিলো। যন্ত্রটি জনসাধারণের মধ্যে এমন আলোড়ন ফেলেছিলো যে স্বয়ং সুইডেনের রানীও একটা পাসকালের যন্ত্রগণক কিনে ফেলে ছিলেন! তবে তখনকার যান্ত্রিক প্রযুক্তিবিদ্যা খুব উন্নত না হওয়ার জন্য পাসকালের যন্ত্রটি খুব নির্ভরযোগ্য করে বানানো সম্ভব হয়নি।

লেবনিজের স্টেপট্ রেকনার



স্টেপট্ রেকনার

পাসকালের যন্ত্রগণক আবিষ্কারের একতিরিশ বছর পরে আর একজন স্বনামধন্য জার্মান গণিতজ্ঞ গডফ্রে লেবনিজ (Gottfried Leibniz) আরো একটু উন্নত ডিজাইনের যন্ত্রগণক ডিজাইন করলেন। এই যন্ত্রটার বৈশিষ্ট্য, এটা কেবল মাত্র যোগই নয়, বিয়োগ, গুণ ভাগ এমন কি বর্গমূলও বার করতে পারতো। যন্ত্রটার নাম দেওয়া হয়েছিল স্টেপট্ রেকনার (Stepped Reckoner)।

স্টেপট্ রেকনার কাজ করতো অনেকটা পাসকালের যন্ত্রের মতো। তবে এতে পাসকালের যন্ত্রের মতো সাধারণ পিন হুইলের বদলে স্টেপট্ হুইল লাগানো হয়েছিল। এটা একটা লম্বাটে সিলিণ্ডার গোছের জনিস, যার গায়ে বিভিন্ন জায়গায় ছোটো ছোটো পিন লাগানো। সিলিণ্ডার ঘোরালে এই পিনগুলো বিভিন্ন গিয়ারের খাঁজে আটকে গিয়ে তাদের ঘোরাতো। ওই গিয়ারগুলোতেই 0 থেকে 9 সংখ্যা লেখা থাকতো আর বসানো থাকতো ছোটো ছোটো জানলার পিছনে। স্টেপট্ হুইলকে কেবলমাত্র ঘোরানোই যেতো না, পাশাপাশি সরানোও যেতা। যখন কোনো সংখ্যা যন্ত্রটাতে ডায়াল করতে হতো তখন স্টেপট্ হুইলকে পাশাপাশি এমনভাবে সরাতে হতো যাতে স্টেপট্ হুইল ঘোরালে যত সংখ্যা ডায়াল করতে হবে, গিয়ারের চাকতিটাও স্টেপট্ হুইলের গায়ে লাগানো ঠিক তত সংখ্যার পিনের সঙ্গে লেগে ঘুরে যায়। অর্থাৎ যদি আমরা 5 সংখ্যাটি ডায়াল করতে চাই তবে স্টেপট্ হুইলকে এমনভাবে সরিয়ে আনতে হবে যাতে স্টেপট্ হুইলে লাগানো অনেক পিনের মধ্য থেকে কেবলমাত্র পাঁচটা পিনের সঙ্গে গিয়ার হুইলের খাঁজ যুক্ত হয়। এবার হুইলটিকে এক পাক ঘোরালেই জানলায় 5 সংখ্যাটা দেখা যাবে। যদি আমাদের 5-এর সঙ্গে 17 গুণ করতে হয় তবে শুধু গুণে গুণে সতেরোবার স্টেপড হুইলের হাতল ঘোরালেই জানলায় গুণ ফলের উত্তর দেখা যাবে।

বিয়োগ করার সময় হাতলকে উল্টো দিকে ঘোরাতে হতো। ভাগ করার সময়ে বার বার বিয়োগ করে ভাগ করতে হতো।

পুরো ব্যাপারটা খুব সহজ মনে হলেও তখনকার ইঞ্জিনিয়ারিং প্রযুক্ত এতই কম উন্নত ছিলো যে পাসকাল বা লেবনিজের এই যন্ত্রগুলো



অ্যানালিটিকাল ইঞ্জিন

খুব একটা নির্ভরযোগ্যতার সঙ্গে কাজ করতো না। তাছাড়া স্টেপট্ রেকনারের হাতলকে গুণে গুণে ঘোরানোর ব্যাপারটাও খুব সুবিধাজনক ছিলো না। তাই আরো উন্নত যন্ত্রগণক তৈরী করার চিন্তা ভাবনা সেখানেই থেমে রইলো না।

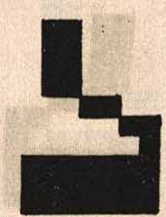
ব্যাবেজের স্বপ্ন অ্যানালিটিকাল ইঞ্জিন

1833 সালে এক ব্রিটিশ গণিতজ্ঞ চার্লস ব্যাবেজ (Charles Babbage) প্রথম যন্ত্রগণকের স্মৃতি ভাণ্ডারের প্রয়োজনীয়তার কথা উপলব্ধি করলেন। তখনও পর্যন্ত যে সব যন্ত্রগণক তৈরী হচ্ছিল, সে সমস্তই যান্ত্রিক গিয়ারের মাধ্যমে গণনা করতো এবং গণনা চালানোর জন্য পরিচালককে গুণে গুণে যন্ত্রে হাতল ঘোরাতে হোতো। ফলে যন্ত্রের দ্রুত অঙ্ক কষার ক্ষমতা পুরোপুরি নির্ভর করতো পরিচালক কতে তাড়াতাড়ি হাতল ঘোরাবেন তার ওপর। ব্যাবেজ বুঝেছিলেন সত্যিকারের কার্যকরী গণনা যন্ত্র তৈরী করতে গেলে প্রথমেই দরকার একটা যান্ত্রিক স্মৃতিভাণ্ডারের ব্যবস্থা করা। এই স্মৃতিতে যে সব অঙ্ক কষার দরকার হবে তার তথ্য প্রথমে প্রবেশ করিয়ে দিলে যন্ত্রগণক তার পরবর্তী কাজকর্ম আরো সহজে করতে পারবে এবং একদম না থেমে পরপর অঙ্কগুলো কষে যেতে পারবে।

ব্যাবেজ আরো ভেবেছিলেন, তাঁর যন্ত্রগণকে হাতল ঘোরানোর বদলে যোগ গুণ ইত্যাদি করার নির্দেশ পাঞ্চ কার্ডের মাধ্যমে দেওয়া হবে। পাঞ্চ কার্ডের ব্যবহার ব্যাবেজের আমলেই শুরু হয়েছিল। এর প্রথম প্রচলন করেন এক ফরাসী যন্ত্রবিদ, নাম জেসফ জ্যাকার্ড। জ্যাকার্ড স্বয়ংক্রিয় তাঁত মেশিনে অটোমেটিক ডিজাইন তোলার জন্য পাঞ্চ কার্ডের ব্যবহার প্রচলন করেন। এখনও তাঁত মেশিনে জ্যাকার্ড প্রবর্তিত পাঞ্চ কার্ডের ব্যবহার হয় এবং এতে বোনা শাড়ি জ্যাকার্ড শাড়ি নামে পরিচিত।

ব্যাবেজের পরিকল্পিত মেশিনের তৃতীয় আর একটা বৈশিষ্ট্য ছিলো। সেটা হলো একই ধরনের অঙ্ক কষার দরকার পড়লে অ্যানালিটিক্যাল মেশিনকে একটা যান্ত্রিক আবর্ত বা লুপ প্রোগ্রামে পরিচালনা করা

যতো যেটা বর্তমানের আধুনিক কম্পিউটর প্রোগ্রামে প্রায়ই ব্যবহার করা হয়।



পিন হুইল

অ্যানালিটিকাল ইঞ্জিনের চতুর্থ বৈশিষ্ট্য থেকে বোঝা যায় ব্যাবেজের অ্যানালিটিকাল মেশিন পরিকল্পনা কতো আধুনিক আর সুদূর প্রসারী ছিলো। ব্যাবেজ অ্যানালিটিকাল ইঞ্জিনের ডিজাইন এমনভাবে করেছিলেন যাতে গণনা হয়ে যাবার পর উত্তরটা ছাপা হয়ে বেরিয়ে আসে!

কিন্তু 1833-এর বৈজ্ঞানিক প্রযুক্তি তখনো ব্যাবেজের পরিকল্পনাকে বাস্তবায়িত করার মতো উন্নত হয়নি। সরকারী অনুদান এবং নিজের ব্যক্তিগত পুঁজি নিঃশেষ করে ব্যাবেজ তখনকার দিনে সতেরো হাজার পাউণ্ড খরচা করেছিলেন তাঁর স্বপ্নের অ্যানালিটিকাল ইঞ্জিনকে রূপদান করতে, কিন্তু শেষ পর্যন্ত তাঁর সে স্বপ্ন স্বপ্নই রয়ে গেল।

তবু ব্যাবেজের এই যুগান্তকরী চিন্তাভাবনা এবং প্রচেষ্টার জন্য তাঁকেই আধুনিক কম্পিউটরের জনক বলা হয়।

পিন হুইল

ব্যাবেজের স্বপ্ন স্বপ্নই রয়ে গেলো, তবে গণনা যন্ত্র বা ক্যালকুলেরের মতো যন্ত্র ক্রমশই উন্নত হতে লাগলো। 1878 সালে ডাবলু. টি. ওডনার একটা খুবই উন্নত যন্ত্রগণক বানালেন, নাম পিন হুইল।

ইলেকট্রিক মটোরের সাহায্যে চালালে পিন হুইল খুবই দ্রুত গণনা করতে পারতো। এখনও যে সব যান্ত্রিক যন্ত্র গণক ব্যবহার করা হয় তা ওই পিন হুইলেরই পরিমার্জিত সংস্করণ।

এর কিছু দিন পরে কম্পটোমিটার নামে আর একটা যন্ত্র তৈর হল। এই যন্ত্রে গণনার তথ্য কী বোর্ডে চাবি টিপে প্রোগ্রাম করা যেতো।



হলেরিথ এবং পানচ্ কার্ড

হলেরিথের পাঞ্চ কার্ড

ডক্টর হারমান হলেরিথ কাজ করতেন অ্যামেরিকান সেনসার অর্থাৎ আদমসুমারি অফিসে। 1887 সালে অ্যামেরিকায় সেনসাস করা হলো। যে কোনো বড় দেশেই সেনসাস হলো এক বিশাল কর্মকাণ্ড। দেশের সঠিক জনসংখ্যা এবং জনগণের সামাজিক ও অর্থনৈতিক অবস্থা ইত্যাদি নিরূপণ করা খুব সহজ ব্যাপার নয়। ব্যাপারটা আরো জটিল হয়ে পড়ে যখন কোটি কোটি মানুষের তথ্য এক জায়গায় জড় করে সঠিক ক্রমে লিপিবদ্ধ করতে হয়।

এই দুরূহ কাজটা উদ্ধার করতে তৎকালীন অ্যামেরিকান সেনসাস ব্যুরো একটি প্রতিযোগিতার কথা ঘোষণা করেন। প্রতিযোগিতার বিষয় ছিলো একটা দ্রুত যন্ত্রগণক তৈরী করা যেটা আদমসুমারির কাজ দ্রুত সম্পন্ন করে দিতে পারবে।

হলেরিথ এরকমই একটা মেশিন বানিয়ে প্রতিযোগিতা জিতে নেন আর মেশিনটার সাহায্যে মাত্র ছয় সপ্তাহের মধ্যেই সমস্ত তথ্য লিপিবদ্ধ করে ফেলা সম্ভব হয়।

হলেরিথ তাঁর মেশিনে ব্যাবেজের পরিকল্পনা মতো পাঞ্চ কার্ড ব্যবহার করেছিলেন। তবে ব্যাবেজ তাঁর পানচ্ কার্ড থেকে তথ্য পড়ার জন্য জ্যাকার্ডের মেশিনের মতো যান্ত্রিক সেনসিং লিভার ব্যবহার করবেন ভেবেছিলেন। সে জায়গায় হলেরিথ ব্যবহার করলেন ইলেকট্রোমেকানিক্যাল ব্যবস্থা। যান্ত্রিক ব্যবস্থার সঙ্গে সঙ্গে বিদ্যুৎ ব্যবহারের ফলে তথ্য পড়ার গতি আরো দ্রুত এবং নির্ভুল হলো।

হলেরিথের ব্যবস্থায় পাঞ্চ কার্ডগুলোকে পারা ভর্তি পাত্রের ওপর রাখা হতো। কার্ডের ওপর সারি সারি অনেকগুলো ধাতব পিন্ থাকতো। তথ্য পড়ার সময়ে ওই পিন্গুলোকে কার্ডের ওপর নামিয়ে আনা হতো। যে পিনের নিচে কার্ডে ফুটো থাকতো সেই পিনটা ফুটোর মধ্যে দিয়ে নিচের পাত্রে রাখা পারার সঙ্গে যুক্ত হয়ে একটা

বৈদ্যুতিক সার্কিট পূর্ণ করতো। আর যে সব পিনের নিচে কার্ডে কোনো ফুটো থাকতো না, সেই সব পিনের বৈদ্যুতিক বর্তনি বা সার্কিট পূর্ণ হতো না। এই ব্যবস্থারই আরো কিছুটা পরিমার্জিত উপায় এখনও কিছু কিছু আধুনিক ইলেকট্রোমেকানিকাল কমপিউটার পাঞ্চ কার্ড রিডার যন্ত্রে ব্যবহার করা হয়।


মার্ক ওয়ান

হলেরিথের মেশিন পরবর্তী চার দশক ধরে খুবই জনপ্রিয় ছিলো। তারপর ইলেকট্রনিক্সের বিকাশের সঙ্গে সঙ্গে এই মেকানিক্যাল গণনা মেশিনের চাহিদাও ফুরিয়ে গেলো। হলেরিথ অবশ্য থেমে ছিলেন না। তিনি 1911 সালে দুটি ভিন্ন কোম্পানির সহযোগিতায় কমপিউটিং ট্যাবুলেটিং রেকর্ডিং কোম্পানি খোলেন। এই কোম্পানিই পরে বিশ্ব বিখ্যাত IBM (International Business Machine) এ রূপান্তরিত হয়। কমপিউটার জগতে I B M-এর অবদান এর গুরুত্ব সর্বাধিক। এরাই এখনও পর্যন্ত পৃথিবীর বৃহতম কম্পিউটার নির্মাতা।

হাওয়ার্ড আইকেনের মার্ক ওয়ান কম্পিউটার

হার্ভাড বিশ্ববিদ্যালয়ের হাওয়ার্ড আইকেন (Howard Aiken) I B M কোম্পানির কাছে প্রস্তাব দেন যে এমন একটা গণনা যন্ত্র বানানো সম্ভব যেটা স্বয়ংক্রিয় ভাবে বিভিন্ন গাণিতিক ক্রিয়াগুলোকে পরপর সাজিয়ে নিতে পারবে।

অসংখ্য বৈদ্যুতিক রিলে আর গোছা গোছা তার দিয়ে তৈরী সেই গণনা যন্ত্রদানব তৈরী হলো 1948 সালে। যন্ত্রটা 1948 সাল থেকে এগারো বছর ধরে সক্রিয় ছিলো। এই সময়ে এর সাহায্যে অনেক জটিল গাণিতিক টেবিল তৈরী করে নেওয়া হয়েছিল। তাছাড়া দ্বিতীয় বিশ্ব মহাযুদ্ধের সময়ে কামানের গোলা, ক্ষেপণাস্ত্র ইত্যাদির গতিপথ অঙ্ক কষে বার করতেও এর সাহায্য নেওয়া হোতো। মার্ক ওয়ান সে সময়ে ছিলো এক বিস্ময়কর বস্তু। কিন্তু তার দ্রুত গণনা ক্ষমতা বৈদ্যুতিক রিলে ব্যবহারের দরুন খুব সীমিত ছিলো। রিলে ছাড়া সে সময়ে আর দ্বিতীয় কোনো উচ্চ গতির সুইচও পাওয়া যেতো না। এই অসুবিধা দূর হলো ইলেকট্রনিক ট্রায়োড আর ডায়োড ভাল্ব আবিষ্কারের সঙ্গে সঙ্গে।



কম্পিউটারের প্রথম প্রজন্ম

প্রথম প্রজন্মের সত্যিকারের কম্পিউটার

ফরাসী বিজ্ঞানী লী ডি ফরেস্ট (Lee De Forest) ট্রায়োড ভাল্ব আবিষ্কার করেন। ট্রায়োড ভাল্বকে নানা কাজে লাগানো যায়। এর সাহায্যে কোনো ক্ষীণ বিদ্যুৎ তরঙ্গকে অনেক গুণ বাড়িয়ে দেওয়া যায়, আবার ডি সি থেকে এ সি বিদ্যুৎও তৈরী করা যায়। এছাড়াও ট্রায়োড ভাল্বের আরো একটা বিরাট গুরুত্বপূর্ণ প্রয়োগ আছে, সেটা হলো এটাকে অত্যন্ত উচ্চ গতি সম্পন্ন সুইচের মতো ব্যবহার করা যায়। ইলেকট্রনিক কমপিউটার নির্মাণের ক্ষেত্রে ট্রায়োডের এই ব্যবহারটাই হলো সব থেকে গুরুত্বপূর্ণ।

পেনসিলভেনিয়া বিশ্ববিদ্যালয়ের দুই গবেষক এই ট্রায়োড ভাল্ব ব্যবহার করে একটা শক্তিশালী কম্পিউটার ডিজাইন করলেন। কমপিউটারটার নাম দেওয়া হলো ENIAC (Electronic Numerical Integrator & Calculator) ENIAC-এর আকৃতিও মার্ক ওয়ানের মতই বিশাল ছিলো। 1946 সালে এটি যখন তৈরী হলো, দেখা গেলো এটি তৈরী করতে উনিশ হাজার ডায়োড আর ট্রায়োড ভাল্ব লেগেছে। রেসিস্টর লেগেছে সত্তর হাজার, সুইচ লেগেছে হাজার ষাটেক আর ক্যাপাসিটর লেগছে দশ হাজার! শুধু তাই নয়, এটি চালানোর জন্য প্রতি ঘণ্টায় এক লাখ পঞ্চাশ হাজার ওয়াট বিদ্যুৎ পুড়তো আর সমস্ত কম্পিউটারে লাগানো অসংখ্য ভাল্বগুলো দুর্গাপুজোর প্যানডেলে লাগানো ইলেকট্রিক আলোর মালার মতো জ্বলতে থাকতো! পুরো কম্পিউটারটা চালু অবস্থায় এতো গরম হতো যে এটাকে পুড়ে যাবার হাত থেকে বাঁচাতে জলের সাহায্যে ঠাণ্ডা করতে হতো।



চিত্র ৫
ভাল্ব

তবে EN(AC) নামে যদানবটি এর পূর্ববর্তী সমস্ত যন্ত্রগণক আর মার্ক ওয়ান কম্পিউটারকে অঙ্ক কষার দ্রুততায় এবং অন্যান্য বৈশিষ্ট্যে একেবারে হারিয়ে দিয়েছিলো। ENIAC তৈরী শেষ হবার আগেই আরো কতোগুলি ইলেকট্রনিক কম্পিউটার ডিজাইন করা হয়। EDSAC (Electronic Delay Storage Automatic Computer), EDVAC (Electronic Discrete Variable Automatic Computer), LEO (Lyons Electronic Office

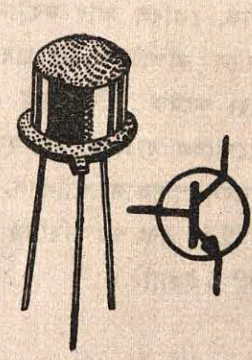
UNIVAC I ইত্যাদি। এই কম্পিউটারগুলোতেও সুইচ্ হিসেবে ট্রায়োড ভাল্ব ব্যবহার করা হয়েছিল। এই ধরনের সমস্ত কম্পিউটারকেই ইদানিং প্রথম প্রজন্মের কম্পিউটার বা First Generation Computers বলে অভিহিত করা হয়। বিভিন্ন ধরনের কম্পিউটারদের এই যে প্রজন্ম হিসেবে ভাগ করার রেওয়াজ, এটা কিন্তু শুরু হয়েছিল IBM কোম্পানির বিজ্ঞাপন থেকে। এই বিজ্ঞাপনে IBM তাঁদের কম্পিউটারকে তৃতীয় প্রজন্মের বলে উল্লেখ করেছিলেন এবং এর পূর্ববর্তী অন্য সমস্ত কম্পিউটারকে প্রথম ও দ্বিতীয় প্রজন্মের বলে চিহ্নিত করেছিলেন।



দ্বিতীয় প্রজন্ম

কম্পিউটারের দ্বিতীয় প্রজন্ম

কম্পিউটার বিজ্ঞানে যখনই কোন একটা গুরুত্বপূর্ণ আবিষ্কার হয়েছে, তখনই পরবর্তী সমস্ত কম্পিউটারেই এই উন্নত প্রযুক্তির প্রয়োগ দেখা গেছে আর এই উন্নত কম্পিউটার গুলিকেই আর একটি নতুন প্রজন্মের কম্পিউটার বলে চিহ্নিত করা হয়েছে। যেমন রিলে সুইচের জায়গায় যখন প্রথম ইলেকট্রনিক সুইচ হিসেবে ট্রায়োড ভাল্ব ব্যবহার করা হলো, তখন ওইধরনের সব কম্পিউটারকে প্রথম প্রজন্মের কম্পিউটার বলে চিহ্নিত করা হলো।



চিত্র ৬ :
ট্রানজিস্টর এবং তার সাংকেতিক চিহ্ন

এর পরে 1947 সালে অ্যামেরিকার বেল ল্যাবরেটরিতে উইলিয়াম শক্‌লে (Willam B. Shokley), জন বার্ডিন (Jhon Bardeen এবং এইচ্ ব্র্যাটেন (H. Brattain) যৌথ ভাবে ট্রানজিস্টর আবিষ্কার করলেন।

ট্রানজিস্টরকেও ট্রায়োডের মতো ইলেকট্রনিক সুইচ্ হিসেবে ব্যবহার করা যায়। ট্রায়োডের থেকে ট্রানজিস্টরের সুবিধে অবশ্য অনেক বেশী। এরা আকারে ছোটো, বিদ্যুৎ খরচ করে কম, বেশী গরম হয় না আর ট্রায়োডের থেকে ঢের বেশী শক্ত সমর্থ। এই ট্রানজিস্টরকে সুইচ্ হিসেবে ব্যবহার করে যে সব কম্পিউটার তৈরী হলো তাদের নাম দেওয়া হলো দ্বিতীয় প্রজন্মের কম্পিউটার।

প্রথম প্রজন্মের থেকে দ্বিতীয় প্রজন্মের কম্পিউটারে অনেকগুলি নতুন বৈশিষ্ট্য দেখা দিলো। প্রথম বৈশিষ্ট্য হলো মডিউলার ডিজাইন



দ্বিতীয় প্রজন্ম

মডিউলার ডিজাইনের সুবিধা হলো, এতে সার্কিটের প্রধান প্রধান অংশগুলিকে আলাদা বোর্ডে তৈরী করা হয়। ফলে কোন একটা প্রধান অংশ খারাপ হলে কেবল সেই অংশটুকুর সারকিট বোর্ড বার করে নতুন বোর্ড লাগিয়ে দিলেই কম্পিউটার ফের চালু হয়ে যায়।

দ্বিতীয় গুরুত্বপূর্ণ বৈশিষ্ট্য হলো, হাই লেভেল ল্যাঙ্গোয়েজ (High Level Language) বা কম্পিউটার প্রোগ্রামিং করার সহজ ভাষার প্রবর্তন। প্রথম প্রজন্মের কম্পিউটার প্রোগ্রামিং করতে হতো যান্ত্রিক ভাষায় (Machine Language)। এই ভাষা সবার পক্ষে চট্ করে শিখে নেওয়া কঠিন ছিলো। 1955 সালে বৈজ্ঞানিক ফর্মুলা এবং অঙ্কের বিভিন্ন সমস্যার সমাধান করার জন্য একটি বিশেষ কম্পিউটার ভাষার উৎপত্তি হলো। ফর্মুলা ট্রান্সস্লেশন (Formula Translation) কথাটাকে সংক্ষিপ্ত করে এই ভাষার নাম রাখা হলো ফরট্রান FORTRAN।

ফরট্রান ভাষা তৈরী হবার পর ক্রমশ আরো বিভিন্ন ধরনের সমস্যা সমাধানের জন্য অন্যান্য কম্পিউটার ভাষা তৈরী হতে লাগলো। যেমন অর্থশাস্ত্র, ব্যাঙ্কের কাজ, টাকা পয়সার হিসেব অর্থাৎ কর্মাসের যাবতীয় কাজকর্ম কম্পিউটারের মাধ্যমে করবার জন্য তৈরী হলো কোবল Cobol)।

তবু সাধারণ কম্পিউটার ব্যবহারকারীদের জন্য ফরট্রান বা কোবল ভাষার থেকেও আরো সহজ ভাষার দরকার ছিলো। 1964 সালে দৈনন্দিন জীবনের নানা রকম কাজকর্ম কম্পিউটারের মাধ্যমে করার জন্য একটা খুব সহজ কম্পিউটার ভাষা তৈরী করা হলো। ভাষাটার নাম বেসিক (Basic)। বেসিক যদিও বিশেষ ধরনের কাজ করার জন্য তৈরী ভাষাগুলির মতো অতো শক্তিশালী নয়, তবু খুব সহজ বলে সারা পৃথিবীতে এটা এখন দারুন জনপ্রিয়।

হাইলেভেল ল্যাংগোয়েজ: শুধু বেসিক, কোবল আর ফরট্রানই নয়, বর্তমানে কম্পিউটর প্রোগ্রাম করতে আরো অনেক গুলি হাই লেভেল ল্যাংগোয়েজ ব্যবহার করা হয় যেমন ডাটাবেস, পাসকাল, অ্যালগল, পি এল 1 (**PLI**) এবং স্বল্প পরিচিত **APL, POP** 2 ইত্যাদি।



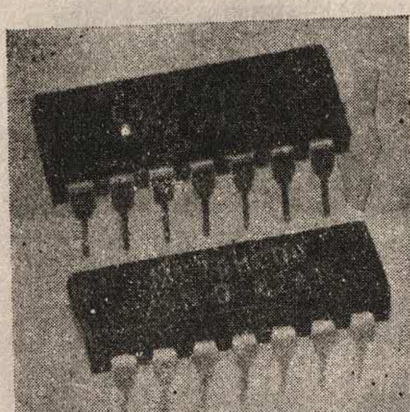
তৃতীয় প্রজন্ম

তৃতীয় প্রজন্মের কম্পিউটার

হাই লেভেল ল্যাংগোয়েজের প্রবর্তনে কম্পিউটর ব্যবহার অনেক সহজ হয়ে গেল। কিন্তু বিজ্ঞানীরা এখানেই সন্তুষ্ট নন, তাঁরা ভাবতে লাগলেন কিভাবে একাধিক ট্রানজিস্টর, রেসিস্টর এবং ক্যাপাসিটর দিয়ে তৈরী একটা ছোটো সার্কিটকে একটা ট্রানজিস্টরের মতো ছোট করে ফেলা যায়। বেশী দিন লাগলো না, রবার্ট নয়েস আর জ্যাক কিলবি (Robert Noyce, Jack Kilby) দুজনে প্রায় একই সময় পৃথক ভাবে এই ক্ষুদ্রকরণের পদ্ধতি আবিষ্কার করলেন। এই পদ্ধতিতে ছোট করে ফেলা সার্কিটকে আই সি বা ইন্টিগ্রেটেড সার্কিট (I C or Integrated Circuit) নাম দেওয়া হলো।

আই সি কথাটা অবশ্য ইদানিং খবর কাগজে টেপ রেকর্ডার আর টেলিভিশনের বিজ্ঞাপনের দৌলতে আমাদের কাছে খুবই পরিচিত। এই আই সিকে কাজে লাগিয়ে তৈরী হলো তৃতীয় প্রজন্মের কম্পিউটার। আই সির ব্যবহারে এই প্রজন্মের কম্পিউটারের আকার অনেক ছোটো হয়ে গেল এবং সেই সঙ্গে বৃদ্ধি পেল কাজকর্মের দ্রুততা। এখনও বিভিন্ন জায়গায় এই তৃতীয় প্রজন্মের কম্পিউটার সক্রিয়ভাবে কাজ করছে। এই প্রজন্মের উল্লেখযোগ্য বৈশিষ্ট্য ছিলো কম দাম, আকারে ছোটো, বিদ্যুৎ শক্তি লাগত কম, বিভিন্ন ধরনের উন্নত

চিত্র ৭ : আই সি



দুটি সাধারণ 14 পিনের ডিজিটাল I C ছবিতে দেখানো হয়েছে। তৃতীয় প্রজন্মের কম্পিউটারে এই ধরনের I C ব্যবহার করা হোত।

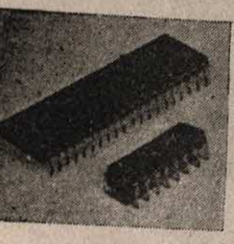
চতুর্থ প্রজন্ম

কম্পিউটার স্মৃতি ব্যবস্থার উদ্ভাবন, যেমন, ম্যাগনেটিক টেপ, হার্ড ডিস্ক ড্রাইভ ইত্যাদি, বিভিন্ন ধরনের প্রোগ্রামিং করার ব্যবস্থা অর্থাৎ ইনপুটের উদ্ভাবন আর আউটপুট হিসেবে VDU এবং হাইস্পিড লাইন প্রিন্টারের ব্যবহার। VDU হলো Video Display Unit কথাটার সংক্ষিপ্ত রূপ। এই ব্যবস্থায় টেলিভিশন স্ক্রিনে কম্পিউটারে পাঠানো নির্দেশ এবং তার উত্তর ফুটিয়ে তোলা হয়। তাছাড়া অত্যন্ত দ্রুত গতিতে কম্পিউটারের নির্দেশ ও উত্তর ছাপার যন্ত্র তৈরী হলো, নাম হাই স্পিড লাইন প্রিন্টার (High Speed Line Printer)। এই প্রিন্টারগুলো এতো দ্রুত ছাপতে পারে যে এতে মিনিটে 200 লাইন পর্যন্ত লেখা ছাপা যায়।

চতুর্থ প্রজন্মের কম্পিউটার

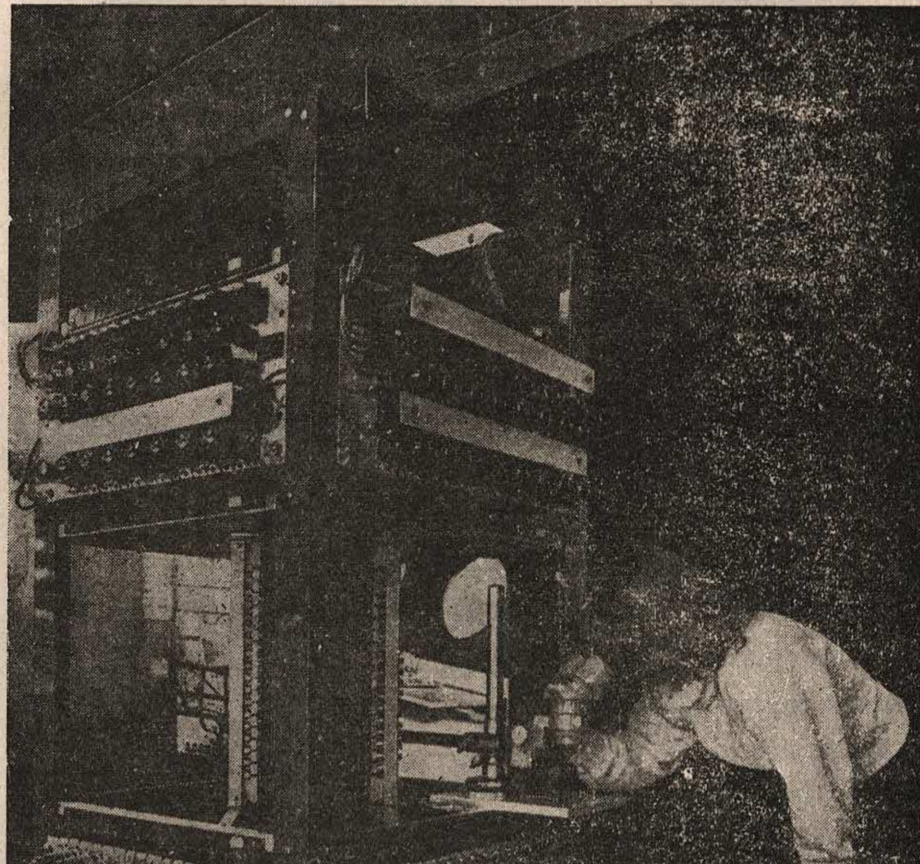
শুধু আই সি তৈরী করেও ইলেকট্রনিক্স বিজ্ঞানীরা সন্তুষ্ট নন্। তাঁরা আই সিরও আবার আই সি তৈরী করতে চাইলেন। মানে এবার আর ছোটো সার্কিট নয়, একটা পুরো বিশাল সার্কিটকেই অতি ক্ষুদ্র করে আই সির মাপে পুরে কেলার জন্য গবেষণা শুরু হলো। শেষ পর্যন্ত দু ইঞ্চি লম্বা, আধ ইঞ্চি চওড়া আর একের আট ইঞ্চি পুরু একটা শুঁয়োপোকার মতো দেখতে প্যাকিং-এর মধ্যে এক লাখের থেকেও বেশী ট্রানজিস্টর আর সেই রকমই সংখ্যা রেসিস্টর আর ক্যাপাসিটর দিয়ে তৈরী একটা প্রকাণ্ড সার্কিট ভরে দেওয়া সম্ভব হলো! এই অতি ক্ষুদ্রায়ণ পদ্ধতির নাম দেওয়া হল VLSI বা ভেরি লার্জ স্কেল ইন্টিগ্রেশন (VeryLarge Scale Integration) টেকনলজি। এর সাহায্যে এখন একটা গোটা কম্পিউটারকে (ইনপুট কী বোর্ড, আউটপুট ডিসপ্লে আর প্রিন্টার সমেত) একটা পকেট ক্যালকুলেটরের মাপে তৈরী করা সম্ভব হচ্ছে। ভাবতে পারেন দ্বিতীয় প্রজন্মের যে যন্ত্রটি আগে দু'দুটো বিরাট ঘরের জায়গা দখল করে নিতো, বিদ্যুৎ শক্তি টানতো একটা ছোটো শহরের মতো, সেই কম্পিউটারের থেকেও বহু গুণে উন্নত চতুর্থ প্রজন্মের কম্পিউটারের সাইজ এখন একটা পেপার ব্যাক পকেট বইয়ের থেকেও অনেক ছোটো। অথচ এতে প্রোগ্রাম করার জন্য টাইপ রাইটারের মতো কী বোর্ড আছে, আউটপুট ডিসপ্লেতে কম্পিউটারের প্রোগ্রাম আর উত্তর ফুটে উঠছে। দরকার পড়লে ওরই মধ্যে যে অতি

চিত্র ৮ :



একটি সাধারণ IC র পাশে একটি VLSI চিপ্ দেখানো হয়েছে। বড় I Cটিই হলো VLSI চিপ্।

চতুর্থ প্রজন্ম



চিত্র ৯ পাশের ছবিতে যে বিরাট যন্ত্রটি দেখা যাচ্ছে সেটি হলে প্রথম প্রজন্মের কম্পউ-টারের একটি 16K বাইট ক্ষমতা সম্পন্ন স্মৃতি ভাণ্ডার। এটি আগাগোড়াই ভাল্ব দিয়ে তৈরী। এই একই ক্ষমতা সম্পন্ন স্মৃতি ভাণ্ডার VSIL টেকনো-লজির দৌলতে এতো ছোট করে ফেলা হয়েছে যে তাকে দেখতে মেয়েটির মাইক্রো স্কোপের সাহায্যে নিতে হচ্ছে।

ছবিটি **National Geographic** পত্রিকা থেকে গৃহীত



ভবিষ্যতের কম্পিউটার

ক্ষুদ্র প্রিন্টার;আছে তাতে সরু একটা কাগজের ফিতের ওপর সমস্ত তথ্যও ছাপিয়ে বেরিয়ে আসছে। এটাকে টিভির সঙ্গে লাগিয়ে দিলে টিভির স্ক্রিনে কম্পিউটারের তথ্য দেখা যাচ্ছে, দরকার পড়লে টেপ রেকর্ডারের সঙ্গে লাগিয়ে বিশেষ তথ্য ক্যাসেটের ফিতেতে রেকর্ড করে রাখা যাচ্ছে। অথচ এতো কাজকর্ম চালাতে কয়েকটা পেন্সিল ব্যাটারিই যথেষ্ট।

এ সমস্তই সম্ভব হয়েছে ওই VLSI টেকনোলজির জন্য। এই ধরনের সার্কিট ব্যবহার করে যে সব কম্পিউটার তৈরী হচ্ছে তাদের বলা হয় চতুর্থ প্রজন্মের কম্পিউটার। VLSI সার্কিটগুলিকে সাধারণভাবে মাইক্রোপ্রসেসর বলা হয়ে থাকে। তাই ছোটো ছোটো চতুর্থ প্রজন্মের কম্পিউটরকে মাইক্রো কম্পিউটার বলা হয়। আমরা বর্তমানে স্কুলে কলেজে যে সব পোর্টেবল টাইপ রাইটারের মতো দেখতে কম্পিউটার দেখছি, ব্যবহার করছি, সেগুলো সবই এই চতুর্থ প্রজন্মের কম্পিউটার।

ভবিষ্যতের পঞ্চম প্রজন্ম

কম্পিউটার টেকনোলজিস্টরা এখনও মোটেই সন্তুষ্ট নন্। তাঁদের বক্তব্য হলো যতদিন না কম্পিউটার তার পরিচালকের সঙ্গে কথা বলতে পারছে আর পরিচালকের গলার আওয়াজ শুনে প্রোগ্রাম নিতে পারছে ততদিন কম্পিউটার টেকনলজির কি আর এমন উন্নতি হয়েছে। কথা বলা কম্পিউটার অবশ্য এখনই তৈরি হয়ে গেছে তবে কথা শুনে প্রোগ্রামিং নেওয়া অথবা কম্পিউটারের আউটপুট হিসেবে গলার স্বরের ব্যবহার এটা এখনও গবেষণাগারের মধ্যেই আবদ্ধ আছে। তাছাড়া এখনও কোনো কম্পিউটারের চিন্তা করার ক্ষমতা কিম্বা কোনো নির্দেশ বলে না দিলে সেটা পরিস্থিতি অনুযায়ি স্থির করে নেওয়ার ক্ষমতা নেই। এটা নিয়েও গবেষণা হচ্ছে কিভাবে কম্পিউটারে কৃত্রিম বুদ্ধি আরোপ করা যায়। আশা করা যায় 1990 সালের পর থেকেই এ ধরনের নানা নতুন প্রযুক্তি এবং উদ্ভাবনের সঙ্গে আমরা পরিচিত হতে থাকবো। এবং সেই সঙ্গে জন্ম নেবে পঞ্চম প্রজন্মের কম্পিউটার।

পরিচ্ছেদ ২

আধুনিক কাম্পউটার

আধুনিক কম্পিউটারের চারটে প্রধান অংশ আছে ; ইনপুট, সেণ্ট্রালপ্রসেসিং ইউনিট, মেমরী আর আউটপুট। কার্যগত দিক থেকে আধুনিক কম্পিউটারের সঙ্গে আমাদের শরীরের অনেক সাদৃশ্য পাওয়া যায়। আমরা যেমন আমাদের পাঁচটি ইন্দ্রিয়ের (চোখ, কান, নাক, জিভ এবং ত্বক) সাহায্যে প্রতি নিয়ত বুঝতে পারি আমাদের চারপাশে কি ঘটছে অর্থাৎ এক কথায় এই পাঁচটি ইন্দ্রিয়ের মাধ্যমেই আমরা আমাদের শরীরের বাইরে ঘটে যাওয়া ঘটনা সম্বন্ধে তথ্য আহরণ করি, ঠিক তেমনি কম্পিউটার তার ইনপুট অংশের মাধ্যমে যে সমস্যা তাকে সমাধান করতে হবে, সেই সমস্যা সম্বন্ধে সমস্ত তথ্য জানতে পারে। আমাদের যেমন একাধিক ইন্দ্রিয়, কম্পিউটারেরও তেমনি বিভিন্ন রকম ইনপুট ব্যবস্থা থাকে। এই সমস্ত ইনপুট ব্যবস্থাকে ইনপুট ডিভাইস বলা হয়। পাঞ্চ কার্ড, পাঞ্চ টেপ, কী বোর্ড ইত্যাদি হলো আধুনিক কম্পিউটারের বিভিন্ন ইনপুট ডিভাইস।

ইনপুট ডিভাইসের মাধ্যমে কম্পিউটারকে কোনো কাজের নির্দেশ দেওয়াকে বলা হয় কম্পিউটার প্রোগ্রামিং (Programing), আর নির্দেশকে বলা হয় প্রোগ্রাম (Program)। কম্পিউটার পরিচালক যিনি এই প্রোগ্রাম করে থাকেন, তাঁকে বলা হয় প্রোগ্রামার (Programar)।

কম্পিউটারের কোনো প্রোগ্রাম করা হলে এই প্রোগ্রাম সর্বপ্রথম কম্পিউটারের অভ্যন্তরিণ স্মৃতি ভাণ্ডারে জমা হয়। প্রোগ্রাম শেষ হবার পর কম্পিউটারের তথ্য বিশ্লেষণী বিভাগ সেণ্টাল প্রসেসিং ইউনিট বা সংক্ষেপে C.P.U এই তথ্য স্মৃতি ভাণ্ডার থেকে সংগ্রহ করে নিয়ে

তাকে বিশ্লেষণ করে দেখে। C.P.U-এর অন্তর্গত একটা ছোট অংশ A.L.U বা এরিথমেটিক লজিক ইউনিট প্রোগ্রামে দেওয়া সমস্ত অঙ্ক কষার কাজ করে। এই পুরো ব্যাপারটার সঙ্গেও কিন্তু আমাদের শরীরের ক্রিয়া কলাপের খুব মিল আছে। আমাদের যখন কোনো প্রশ্ন করা হয় তখন সেই প্রশ্ন আমাদের মস্তিষ্কের স্মৃতি ভাণ্ডারে জমা হয়। পুরো প্রশ্নটা তারপর মস্তিষ্ক বিশ্লেষণ করে দেখে উত্তরটা আমাদের জানা আছে কিনা। যদি জানা থাকে তবে আমরা সেটা বলে দিই কিম্বা কাগজে লিখে দিই।

আধুনিক কম্পিউটারেও এই লিখে দেওয়া এবং বলে দেওয়ার ব্যবস্থা রাখা হয়েছে। C.P.U যখনই পুরো তথ্য বিশ্লেষণ করে তার উত্তর বার করে ফেলে তখনই (পূর্বে দেওয়া নির্দেশ অনুসারে) সেই উত্তর টিভির মতো একটা যন্ত্রের স্ক্রিণে ফুটে ওঠে। এই যন্ত্রের নাম কম্পিউটারের মনিটর বা V.D.U (ভিডিও ডিসপ্লে ইউনিট)। এছাড়া ওই একই উত্তর কম্পিউটার পরিচালিত একটা ছোট্ট লেখা ছাপানো যন্ত্রের (Printer) মাধ্যমে কাগজে ছেপে বেরিয়ে আসে। এই কাগজে ছাপানো উত্তরকে কম্পিউটার পরিভাষায় হার্ড কপি (Hard Copy) বলা হয়।

যে সব যন্ত্রের মাধ্যমে কম্পিউটার তার উত্তর বলে দেয় সেই সব যন্ত্রকে এক কথায় আউটপুট ডিভাইস বলা হয়। ইদানিং আউটপুট ডিভাইস হিসেবে স্পিচ্ প্রসেসর ইলেক্ট্রনিক সার্কিট ব্যবহার হচ্ছে। এই ব্যবস্থার সাহায্যে কম্পিউটার মানুষের গলার স্বরে উত্তর জানাতে পারে অথবা প্রোগ্রামারের দেওয়া নির্দেশ অনুসারে গানের সুর তৈরী করতে পারে।

সব শেষে আসে আমাদের শরীরের হাত পা। আধুনিক কম্পিউটারেরও কিন্তু হাত পা আছে আর তাকেই বলে রোবট। রোবট হলো কম্পিউটার নিয়ন্ত্রিত একটা জটিল যান্ত্রিক ব্যবস্থা যাতে চলা ফেরার জন্য ইলেকট্রনিক মটোর চালিত চাকা থাকে এবং হাইড্রলিক ব্যবস্থার সাহায্যে নিয়ন্ত্রিত নানা রকম যান্ত্রিক হাত থাকে। রোবটের চলা ফেরা, যান্ত্রিক হাতের নড়াচড়া সবই নিয়ন্ত্রণ করে কম্পিউটারের C.P.U এবং স্মৃতি ভাণ্ডার। বর্তমানে বিদেশের গাড়ির কারখানায় রোবটের



আধুনিক কম্পিউটার

চিত্র ১০ :

ছবিতে ফিলিপস্ কম্পানির তৈরী স্পিচ্ সিন্থেসাইজার লাগানো কম্পিউটার দেখা যাচ্ছে। এই কম্পিউটার মানুষের গলার উত্তর বলে দেয়।

ব্যবহার তো ভীষণ ভাবে বেড়ে গেছে। যন্ত্রাংশ জুড়ে গাড়ি তৈরী করা, তাকে রং করা, সব কাজই রোবট করছে। তাছাড়া বাচ্চাদের খেলার জন্য এবং বাড়ি পাহারা দেওয়ার জন্যও ছোট ছোট রোবট বিদেশে কিনতে পাওয়া যায়। এ ধরনের রোবটে এমন প্রোগ্রাম করা থাকে যে এটা সব সময় বাচ্চার পিছনে পিছনে ঘরের মধ্যে ঘোরে (ঠিক পোষা কুকুরের মতো)। রোবটকে যদি বলেন কবিতা বলো, সে কবিতা বলে। ভোরবেলা গান গেয়ে ঘুম ভাঙায়। রাত্রিবেলা একা একা সমস্ত ঘরে ঘুরে ঘুরে পাহারা দেয় আর চোর ঢুকলেই অ্যালার্ম বাজায়!

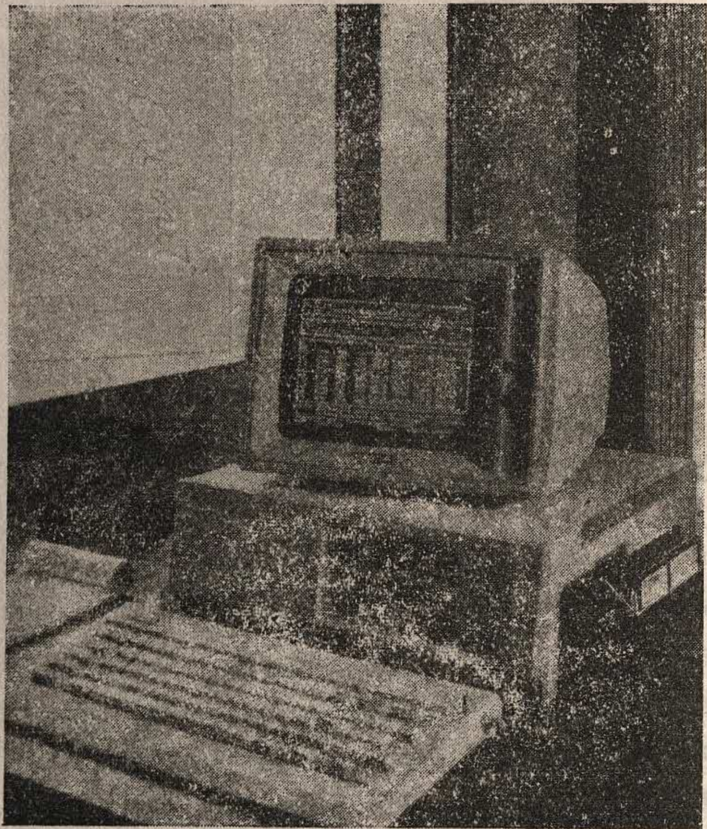
তবে কল্প বিজ্ঞানের সিনেমাতে যেমন মানুষের মতো দেখতে রোবট দেখা যায়, আধুনিক রোবট কিন্তু মোটেই মানুষের মতো দেখতে নয়। উপরন্তু তার নিজের চিন্তা করার ক্ষমতা একদমই নেই। যে নির্দেশ তাতে ভরা আছে সে ঠিক সেই কাজই করে, অন্য নির্দেশ একেবারেই বোঝে না।

আধুনিক মাইক্রো কম্পিউটার কেমন দেখতে হয়

আমরা যে সব মাইক্রো কম্পিউটার নিয়ে স্কুলে বা কলেজে কাজ করবো সেগুলো সবই ছোটো পোর্টেবল টাইপ রাইটারের মতো দেখতে। টাইপ রাইটারের মতই তাতে কী বোর্ড থাকে। শুধু যে

২

আধুনিক কম্পিউটর



চিত্র ১১ ॥ একটি আধুনিক পারশোনাল মাইক্রো কম্পিউটার।
1—কী বোর্ড
2—মূল কম্পিউটার এবং ফ্লপি ড্রাইভ
3—মনিটর
4—ফ্লপি ডিস্ক্।

জিনিসটা থাকেনা, সেটা হলো টাইপ করার কাগজ লাগানোর ব্যবস্থা। এই যন্ত্রটিই হলো মূল কম্পিউটার। এর সঙ্গে আরো অনেক ধরনের সাহায্যকারী পেরিফেরিয়ালস্ (Peripherials) যন্ত্র লাগানো যায়। যার মধ্যে সব সময়ই যেটি লাগানো থাকে সেটি হলো VDU (Video Display Unit)। একে অনেক সময় মনিটরও বলা হয়। VDU আসলে একটা টিউনার এবং অডিও অ্যামপ্লিফায়ার ছাড়া টিভি সেট। এতে দূরদর্শণের প্রোগ্রাম ধরা যায় না কিম্বা শব্দও শোনা যায় না। এর সঙ্গে কম্পিউটরকে লাগালে, স্ক্রিনে কম্পিউটরে পাঠানো নির্দেশ এবং কম্পিউটরের উত্তর দুটোই ফুটে ওঠে। VDU কালো সাদা কিম্বা রঙিন দু ধরনেরই হতে পারে। যে সব কম্পিউটরে রঙিন অক্ষর কিম্বা রেখা ফুটিয়ে তোলার ব্যবস্থা আছে তাতে রঙিন VDU

লাগালে ভাল হয়। ইদানীং বেশীর ভাগ ছোটো মাইক্রো কম্পিউটারের (যাকে আবার পারশোনাল কম্পিউটার অর্থাৎ ব্যক্তিগত কম্পিউটার বলা হয়ে থাকে) ভেতরে UHF to VHF মডিউলেটরের ব্যবস্থা থাকে। এর সাহায্যে কম্পিউটারকে বাড়ির সাধারণ টিভি সেটের অ্যানটেনার সকেটে লাগিয়ে দিলে VDU-র মতো টিভির স্ক্রিনে লেখা ফুটে ওঠে।



আধুনিক কম্পিউটার

VDU, কী-বোর্ড এবং স্মৃতি ব্যবস্থা ছাড়া মাইক্রো কম্পিউটারে আর যে পেরিফেরাল যন্ত্র খুব বেশী ব্যবহার করা হয়, সেটি হলো প্রিনটার। যদিও কম্পিউটারের উত্তর ও ফলাফল সবই VDU বা টিভি স্ক্রিনে দেখতে পাওয়া যায়, তবুও, সেগুলি যদি ছাপা হয়ে বেরিয়ে আসে তবে সেটা কার না ভালো লাগে ? এই জন্য কম্পিউটারের সঙ্গে লাগানো হয় প্রিনটার।

আধুনিক ইলেকট্রনিক টাইপ রাইটারকেও বিশেষ সংযোগ ব্যবস্থার মাধ্যমে (এমনকি কোন কোন মডেলকে একেবারে সরাসরি) কম্পিউটারের সঙ্গে প্রিনটার হিসেবে লাগানো যায়। প্রিনটার হিসেবে টাইপরাইটারকে ব্যবহার করলে তখন কিন্তু সেটাতে আর টাইপ করতে হয় না। কম্পিউটারের কী বোর্ডে যা টাইপ করা হচ্ছে এবং কম্পিউটারের যা ফলাফল VDU-তে দেখা যাচ্ছে, সবই ওই টাইপরাইটারে অটোমেটিক ছাপা হয়ে বেরিয়ে আসতে থাকে।

এগুলি ছাড়া আরো অনেক ধরণের পেরিফেরালস্ আছে যেমন গ্রাফিক ট্যাবলেট, লাইট পেন, ডিজিটাল জয়স্টিক ইণ্টারফেস, মডেম্ ইত্যাদি। তবে এদের ব্যবহার এতো ব্যাপক নয়।

চিত্র ১২



মডেম (Modem)। এর সাহায্যে কম্পিউটারের তথ্য বা প্রোগ্রাম সাধারণ টেলিফোন লাইনের মাধ্যমে দূর, দূরান্তরে পাঠানো যেতে পারে।

ইনপুট এবং আউটপুট ডিভাইস ছাড়া কম্পিউটারের সঙ্গে আরো একটি বাড়তি সংযুক্তি লাগানো হয়, সেটি হলো বাড়তি স্মৃতি ভাণ্ডার। এটির কাজ হলো দরকারী প্রোগ্রাম বা তথ্যগুলিকে স্থায়ীভাবে রেকর্ড করে রাখা যাতে কম্পিউটার অফ্ করে দিলেও রেকর্ড করা প্রোগ্রামটি মুছে না যায়। ছোট পকেট কম্পিউটারে RAM (Random Access Memory) কিম্বা DRAM (Dynamic RAM) ধরনের LSI সার্কিট দিয়ে তৈরী ইলেকট্রনিক স্মৃতি, কম্পিউটারের বাড়তি স্মৃতি হিসাবে ব্যবহার করা হয়। আর মাইক্রো কম্পিউটারে 45 RPM ইপি রেকর্ডের মতো দেখতে ফ্লপি ডিস্ক (Floppy Disc) নামে একরকম চৌম্বক ডিস্কে বাড়তি স্মৃতি টেপ করে রাখা হয়। এছাড়াও আরো উন্নত ধরনের তথ্য ও স্মৃতি সংরক্ষণ ব্যবস্থা আছে, যেমন ম্যাগনেটিক টেপ, উইনচেস্টার ড্রাইভ ইত্যাদি। সাধারণত বড় ধরনের অত্যন্ত দ্রুত আর দামী কম্পিউটারেই উইনচেস্টার হার্ড ডিস্ক কিম্বা ম্যাগনেটিক টেপ স্মৃতি সংরক্ষণ করতে ব্যবহার করা হয়। মাইক্রো কম্পিউটারের বাড়তি স্মৃতি হিসেবে ফ্লপি এবং ডিজিটাল ক্যাসেট রেকর্ডারের প্রচলন বেশী। এছাড়া বেশ কয়েকটি জনপ্রিয় মাইক্রো কম্পিউটারের অভ্যন্তরে এমন ব্যবস্থা করা থাকে, যার সাহায্যে সাধারণ গান শোনার ক্যাসেটেই কম্পিউটার প্রোগ্রাম রেকর্ড করা যায়।

□

পরিচ্ছেদ ২

বিভিন্ন সংখ্যা পদ্ধতি

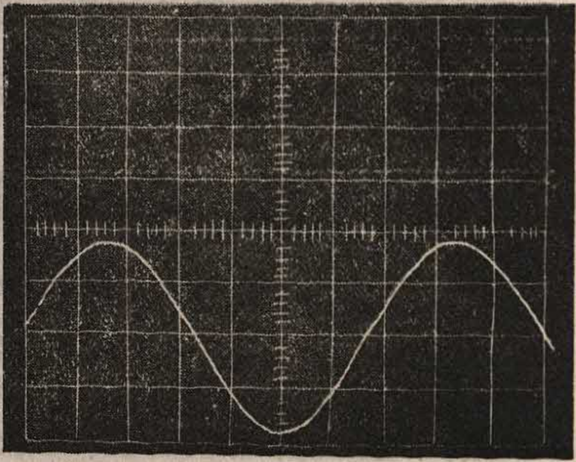
অ্যানালগ এবং ডিজিটাল সংকেত

আমাদের সঙ্গে কম্পিউটারের মূল তফাৎ হলো, আমরা যেমন অ, আ, ক, খ, ১, ২, ৩ ইত্যাদি হরফগুলি তাদের আলাদা আলাদা চেহারা দেখে চিনতে পারি, কম্পিউটার কিন্তু তা পারে না। প্রতিটা হরফ এবং সংখ্যাকে কম্পিউটার আলাদা আলাদা বিশেষ একসারি বিদ্যুৎ স্পন্দনের সাহায্যে চিনে নেয়।

আমরা রেডিও বা স্টিরিওতে যে গান শুনি সেটার উৎপত্তি স্পিকারের কাগজের কম্পন থেকে। এই কম্পন সৃষ্টি করা হয় স্পিকারের কয়েলের মধ্য দিয়ে বিশেষ বিদ্যুৎ স্পন্দন পাঠিয়ে। স্পিকার কয়েলের মধ্য দিয়ে প্রবাহিত এই বিদ্যুৎ স্পন্দন, এবং কমপিউটারে যে বিদ্যুৎ স্পন্দন ব্যবহার করা হয়, সে দুটি সম্পূর্ণ আলাদা ধরণের। গান বাজনা, টিভি, টেলিফোন ইত্যাদিতে যে বিদ্যুৎ স্পন্দন ব্যবহার করা হয়, তাকে যদি অসিলোস্কোপ যন্ত্রে দেখা যায়, তবে দেখা যাবে একটা তরঙ্গায়িত লাইনের ছবি। লাইনটার তরঙ্গগুলি গান বা বাজনার তালে তালে নানা রকম ভাবে পাল্টাতে থাকবে। কখনো সেগুলিকে করাতের দাঁতের মতো দেখাবে, কখনো বা চিরুনির দাঁড়ার মতো, কখনো বা স্প্রিংএর মতো আবার কখনো ঢেউ এর মতো। আমরা যারা বিজ্ঞান জানি তারা অসিলোস্কোপের এই ছবির দিকে তাকিয়েই বলে দিতে পারবো এটা হলো অ্যানালগ সিগন্যালের ছবি। অ্যানালগ (Analog) সিগন্যালের বৈশিষ্ট্য হলো এই বিদ্যুৎ স্পন্দনের ভোল্টেজ, নেগেটিভ থেকে পজেটিভ তরঙ্গ উচ্চতা (amplitude) পর্যন্ত যে কোনো মানে

থাকতে পারে। তাছাড়া তরঙ্গের পজেটিভ থেকে নেগেটিভ বা নেগেটিভ থেকে পজেটিভ তরঙ্গ উচ্চতায় পরিবর্তনের হারও যা খুশি হতে পারে। এই দুটি বৈশিষ্ট্যের জন্য অ্যানালগ সিগন্যালের চেহারা নানারকম বিচিত্র ধরণের হয়। হার্টের ই. সি. জির গ্রাফ দেখেছেন? এটি হলো

অ্যানালগ এবং ডিজিটাল সংকেত



চিত্র ১৩

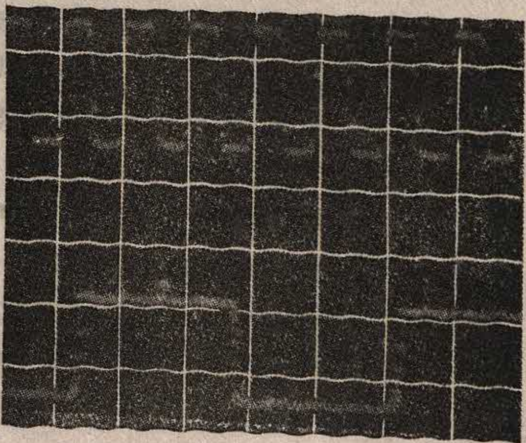
অসিলোস্কোপ স্ক্রিনে অ্যানালগ তরঙ্গের ছবি

অ্যানালগ সিগন্যালের একটি চমৎকার উদাহরণ।

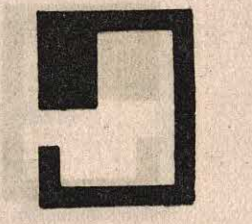
এবার যদি একটি কমপিউটারের ইনপুট ডিভাইসকে অসিলোস্কোপের সঙ্গে লাগিয়ে চালানো হয় তবে দেখা যাবে একটা অন্যরকম ছবি। ছবিটা হবে অসিলোস্কোপের স্ক্রিনে সমান্তরাল দুটি আনুভূমিক (Horizontal) লাইন বরাবর ছোটো ছোটো ড্যাশ চিহ্নের মতো দাগ। উপরের লাইনে যখন ড্যাশ দাগ তখন তার ঠিক নিচের লাইনে কোনো দাগ নেই, আবার নিচের লাইনে যখন দাগ, তখন ঠিক ওপরের

চিত্র ১৪

আসিলোস্কোপ স্ক্রিনে ডিজিটাল তরঙ্গের ছবি



লাইনে দাগ নেই। প্রথম দর্শনে মনে হতে পারে এ আবার কি? কিন্তু আমরা যারা বিজ্ঞান জানি, তারা সঙ্গে সঙ্গে বলবো, এটি হলো স্কোয়্যার ওয়েভের (Square wave) ছবি আর সিগন্যালটি ডিজিটাল সিগন্যাল (Digital Signal)। স্কোয়্যার ওয়েভের বৈশিষ্ট্য হলো, এতে পজেটিভ আর নেগেটিভ তরঙ্গ উচ্চতা দুটি নির্দিষ্ট। ফলে এই সিগন্যালে মাত্র দুটি ভোল্টেজ পাওয়া যায়। একটি পজেটিভ তরঙ্গ উচ্চতার ভোল্টেজ আর অন্যটি নেগেটিভ তরঙ্গ উচ্চতার ভোল্টেজ। এছাড়া অন্য কোনো রকম ভোল্টেজ ডিজিটাল সিগন্যালে পাওয়া যায় না। এই বৈশিষ্ট্যের ফলে ডিজিটাল সিগন্যালের ভোল্টেজ অত্যন্ত দ্রুত পজেটিভ থেকে নেগেটিভে এবং নেগেটিভ থেকে পজেটিভে পরিবর্তিত হয়। সেইজন্য অসিলোস্কোপ স্ক্রিনে পজেটিভ আর নেগেটিভ তরঙ্গ-উচ্চতা দুটি আলাদা সমান্তরাল লাইন হিসেবে দেখা যায়। পজেটিভ থেকে নেগেটিভ বা নেগেটিভ থেকে পজেটিভের মধ্যে পরিবর্তন এতই দ্রুত হয় যে বেশীর ভাগ অসিলো-স্কোপেই উপর আর নিচের ড্যাশচিহ্ন-সংযোগকারী আলোর রেখা আর চোখেই পড়ে না।



অ্যানালগ এবং ডিজিটাল সংকেত

ডিজিটাল সিগন্যালই কম্পিউটারের আসল ভাষা, যার সাহায্যে কম্পিউটার কোনো সংখ্যা, হরফ বা নির্দেশ বোঝে। এই ভাষাতেই কম্পিউটার হিসেব করে, উত্তর বলে। কিন্তু আমরা আবার কম্পিউটারের ডিজিটাল ভাষা চট্ করে বুঝতে পারি না। সেইজন্য দরকার হয় ইনপুট আর আউটপুট ডিভাইসের। ইনপুট ডিভাইসের সাহায্যে আমরা আমাদের পরিচিত হরফ এবং সংখ্যায় কমপিউটারে তথ্য পাঠাই। ইনপুট ডিভাইস আমাদের অজান্তে এই তথ্যের প্রতিটা হরফ, সংখ্যা বা নির্দেশকে কমপিউটারের বোধ্য ডিজিটাল সিগন্যালে পরিবর্তিত করে দেয়। ঠিক তেমনি আউটপুট ডিভাইস যন্ত্রের মাধ্যমে কমপিউটার থেকে বেরিয়ে আসা ডিজিটাল সিগন্যালকে আমাদের বোধ্য হরফ বা সংখ্যায় টিভি স্ক্রিনে ফুটিয়ে তোলা হয়, কিম্বা কাগজে ছাপিয়ে নেওয়া হয়।

কম্পিউটার বুঝতে গেলে তাই প্রথমেই কম্পিউটারের ডিজিটাল ভাষা বোঝা একান্ত দরকার। ডিজিটাল ভাষা বোঝার সুবিধার জন্য ডিজিটাল সিগন্যালকে কাগজে কলমে গাণিতিক চিহ্নের মাধ্যমে লেখা হয়। এই ব্যাপারে আন্তর্জাতিক নিয়ম হলো, ডিজিটাল সিগন্যালের স্কোয়ার ওয়েভ তরঙ্গের পজেটিভ অ্যামপ্লিচিউডকে আমরা

1 সংখ্যার দ্বারা বোঝাবো এবং নেগেটিভ অ্যামপ্লিচিউডকে আমরা 0 সংখ্যা দিয়ে বোঝাবো। একটা কথা এখানে মনে রাখা দরকার, সেটা হলো এই 1 এবং 0 আসলে ডিজিটাল সিগন্যালের পজেটিভ আর নেগেটিভ অ্যামপ্লিচিউডের প্রতিনিধি বা রূপক। একে আমাদের পরিচিত 1 এবং 0 সংখ্যার সঙ্গে গুলিয়ে ফেললে চলবেনা।

ডিজিটাল সিগন্যালে যেহেতু দুটো মোটে ভোল্টেজ বা স্টেট (state) আছে তাই ডিজিটাল সিগন্যালের লেখ্য রূপেও দুটো সংখ্যা আছে, 1 এবং 0। এই 1 এবং 0 দিয়ে গঠিত কম্পিউটার ভাষাকে বাইনারি নাম্বার (Binary number) বলা হয়। আমরা অবশ্য অঙ্ক কষার সময় ব্যবহার করি দশমিক বা ডেসিম্যাল নাম্বার (Decimal number)। তাই দশমিক সংখ্যার অঙ্ক কম্পিউটারকে বোঝাতে গেলে প্রতিটা ডেসিম্যাল নাম্বারের জন্য আলাদা আলাদা বাইনারি নাম্বার তৈরী করতে হয়। সুখের কথা হলো কোন্ ডেসিম্যাল নাম্বারের বাইনারি কত বা কোন্ বাইনারি নাম্বারের ডেসিম্যাল নাম্বারটি কতো সেটার জন্য আমাদের কোনো টেবিল বা চার্ট মুখস্ত করতে হয় না। ডেসিম্যাল নাম্বার থেকে বাইনারি নাম্বার বা বাইনারি নাম্বার থেকে ডেসিম্যাল নাম্বার তৈরী করার (Conversion) জন্য অঙ্কের সহজ ফর্মুলা আছে। কিন্তু ইংরেজি হরফ এবং কিছু বিশেষ চিহ্ন (Special Character) যেমন : ; + / ইত্যাদি এবং কমপ্পিউটারের বিভিন্ন রকম কাজ করার নির্দেশ (Command), এদেরকে বাইনারি নাম্বারে পরিবর্তন করার কোনো ফর্মুলা নেই।

অ্যাসকি কোড

প্রথমদিকে কম্পিউটার নির্মাতারা কম্পিউটার তৈরী করার সময় তাঁদের সুবিধা মতো বিভিন্ন হরফ, স্পেশাল ক্যারেকটার এবং কম্যাণ্ডের (Command) নিজস্ব বাইনারি নাম্বার চার্ট তৈরী করে নিতেন। এর ফলে কিছুদিনের মধ্যেই বিরাট অসুবিধা দেখা দিল। কোন এক কম্পানীর কম্পিউটারে যদি অন্য কম্পানীর ইনপুট যন্ত্র বা আউটপুট যন্ত্র লাগানো হয় তবে সেটি আর কাজ করে না। অর্থাৎ যদি সমস্ত টেপ রেকর্ডার নির্মাতা তাঁদের টেপ বাজানোর জন্য আলাদা আলাদা টেপ স্পিড ব্যবহার করতেন তবে যে কাণ্ডটি হতো সে রকম আর কি! পরে এই অসুবিধা দূর করার জন্য একটি আন্তর্জাতিক কোড তৈরী

হলো, যার নাম আমেরিকান স্ট্যানডার্ড কোড ফর ইনফরমেশন ইণ্টারচেঞ্জ (American Standerd Code for Information Interchange) যাকে সংক্ষিপ্ত ভাবে ASCII (উচ্চারণ অ্যাসকি) কোড বলা হয়। বর্তমানে সমস্ত কম্পিউটারের ইনপুট আর আউটপুট যন্ত্রগুলি অ্যাসকি কোড অনুযায়ী তৈরী করা হচ্ছে। তাই এখন যে কোনো কমপিউটার ক্রেতাই তাঁর ইচ্ছেমতো সস্তা দেখে, অথবা বিশেষ প্রয়োজন অনুযায়ী বিভিন্ন কম্পানির তৈরী কমপিউটারের বিভিন্ন অংশ এক সঙ্গে জুড়ে, তাঁর নিজস্ব কম্পিউটার সিসটেমটি তৈরী করে নিতে পারছেন।

অ্যাসকি কোড

সারণী ১

অ্যাসকি কোডের সারণী

	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	\	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	S_0	(	8	H	X	h	x
1001	HT	S_1	)	9	I	Y	i	y
1010	LF	S_2	*	:	J	Z	j	z
1011	VT	S_3	+	;	K	[	k	{
1100	FF	S_4	,	<	L	\	l	|
1101	CR	S_5	—	=	M	]	m	}
1110	SO	S_6	.	>	N	↑	n	-
1111	S1	S_7	/	?	O	←	o	DEL

A= 100 0001

\= 110 0000

দশমিক সংখ্যা পদ্ধতি

দশমিক সংখ্যা পদ্ধতি

আমরা যে সংখ্যায় অঙ্ক কষি সেটি হলো দশমিক বা ডেসিম্যাল নাম্বার। এতে মোট দশটি সংখ্যা (0 থেকে 9) আছে আর এই দশটি সংখ্যাকে বিভিন্ন ভাবে সাজিয়ে আমরা সমস্ত সম্ভাব্য সংখ্যা লিখতে পারি। যদি ধরা যায় 19637 সংখ্যাটি, এটি লিখতে আমরা 1, 9, 6, 3, 7, ব্যবহার করছি। এবার একটু চিন্তা করে দেখুন, 19637 সংখ্যাটিতে একক, দশক, শতক ইত্যাদি জায়গায় যত সংখ্যা আছে, ততো সংখ্যক একক, দশক, শতক, সহস্র সংখ্যা যোগ করলেই 19637 সংখ্যাটি পাওয়া যায়।

অর্থাৎ $19637=(1\times10^4)+(9\times10^3)+(6\times10^2)+(3\times10^1)+(7\times10^0)$

$=(1\times10000)+(9\times1000)+(6\times100)+(3\times10)+(7\times1)$

$=10000+9000+600+30+7$

$=19637$

তাহলে দেখা যাচ্ছে দশমিক পদ্ধতিতে 10 সংখ্যাটিই হচ্ছে র‍্যাডিক্স (Radix)। অঙ্কের ভাষায় কোন একটি গাণিতিক সংখ্যা পদ্ধতির (number system) র‍্যাডিক্স বলতে বোঝায় এমন একটি সংখ্যা যার 0, 1, 2, 3, 4, 5 ইত্যাদি Power গুলি হলো সেই সংখ্যা পদ্ধতির পরপর ছোট থেকে বড় Position value ; ব্যাপারটা একটু গুলিয়ে গেল না ? একটা উদাহরণ দিয়ে বোঝা যাক—

আমাদের এই দশমিক পদ্ধতির কথাই ধরুন না। এর র‍্যাডিক্স হলো 10। কেন ? কারণ, দশমিক সংখ্যায় পরপর ছোট থেকে বড় Position value গুলি হলো একক, দশক, শতক, সহস্র ইত্যাদি। এই বার দেখুন এই 10 এরই 0, 1, 2, 3, Power গুলিই হলো একক, দশক, শতক ইত্যাদি সংখ্যা

$10^0 = 1$ (একক)

$10^1 = 10$ (দশক)

$10^2 = 100$ (শতক)

$10^3 = 1000$ (সহস্র)

সুতরাং কেবল মাত্র দশ এর বিভিন্ন Power কে প্রয়োজন মতো যোগ করে দশমিক সংখ্যা পদ্ধতির সমস্ত সংখ্যা লেখাসম্ভব। উদাহরণ

হিসাবে দেখুন কিভাবে 10 এর বিভিন্ন Power যোগ করে আমরা দশমিক সংখ্যা পদ্ধতির বিভিন্ন সংখ্যা গুলি লিখিতে পারি।

$1=10^0$, $2=10^0+10^0=1+1$, $5=10^0+10^0+10^0+10^0+10^0=1+1+1+1+1$, $22=10^1+10^1+10^0+10^0=10+10+1+1$ ইত্যাদি।

বাইনারি সংখ্যা পদ্ধতি

কম্পিউটার অঙ্ক কষে বাইনারি নাম্বার বা বাইনারি সংখ্যা পদ্ধতিতে। আমরা একটু আগেই জেনেছি বাইনারি সংখ্যা পদ্ধতিতে মোটে দুটি চিহ্ন বা Symbol আছে। সে দুটি হলো 0 এবং 1. কেবল মাত্র এই দুটি সংখ্যাকে বিভিন্ন ভাবে সাজিয়ে আমরা বাইনারি পদ্ধতির সমস্ত রকম সংখ্যা লিখতে পারি। যেমন 0, 1, 10, 11, 101, 11101 ইত্যাদি।

বাইনারি সংখ্যা পদ্ধতির Radix হলো 2. 2-এর 0, 1, 2, 3 ইত্যাদি Power গুলি বাইনারি সংখ্যা পদ্ধতির ছোট থেকে বড় Position Value নির্দেশ করে। বাইনারি সংখ্যা পদ্ধতিতে পর পর Position Value গুলি হলো—

$2^0=1$

$2^1=2$

$2^2=4$

$2^3=8$

$2^4=16$

বাইনারি থেকে দশমিকে পরিবর্তন

উদাহরণ হিসেবে বাইনারি 11101 সংখ্যাটিই ধরা যাক না কেন। এই সংখ্যাটি দেখলে আমাদের প্রথমেই মনে হবে এগারো হাজার একশো এক সংখ্যার কথা। কিন্তু সে তো হবে দশমিক পদ্ধতিতে। বাইনারি পদ্ধতির Radix আর দশমিক পদ্ধতির Radix আলাদা। তাই বাইনারি 11101 সংখ্যাটির সমান দশমিক সংখ্যার মান হবে আলাদা।

আসুন, এবার দেখা যাক বাইনারি 11101 সংখ্যাটি ডেসিম্যাল সংখ্যা পদ্ধতির কোন সংখ্যা নির্দেশ করে।

বাইনারি সংখ্যা পদ্ধতি

পদ্ধতিটি সেই দশমিক সংখ্যারই অনুরূপ। দশমিক 19637 কে তার Radix এর সাহায্যে আমরা যে ভাবে ভেঙে লিখেছিলাম, সে ভাবেই 11101কে তার র‍্যাডিক্সের সাহায্যে ভেঙে লিখতে হবে। 19637 কে আমরা ভেঙে লিখেছিলাম—

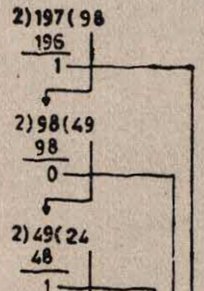
$$(1\times10^4)+(9\times10^3)+(6\times10^2)+(3\times10^1)+(7\times10^0)$$

ঠিক তেমনি 11101 কে আমরা লিখবো—

$$11101=(1\times2^4)+(1\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$$
$$=(1\times16)+(1\times8)+(1\times4)+(0\times2)+(1\times1)$$
$$=16+8+4+1$$
$$=29$$

সুতরাং বাইনারি 11101 সংখ্যাটি, দশমিক 29 সংখ্যাকে বোঝায়। আর একটি উদাহরণ দেখা যাক ঃ—

$$11000101=(1\times2^7)+(1\times2^6)+(0\times2^5)+(0\times2^4)+(0\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$$
$$=128+64+0+0+0+4+0+1$$
$$=197\ (\text{দশমিক})$$



চিত্র ১৫

দশমিক থেকে বাইনারিতে পরিবর্তন

তাহলে দেখা যাচ্ছে বাইনারি সংখ্যা আর দশমিক সংখ্যার মধ্যে একটি নির্দিষ্ট সম্পর্ক আছে। নিচে দশমিক 0 থেকে 9 পর্যন্ত সংখ্যার বাইনারি কত হবে তার একটি টেবিল দেওয়া হলো। কয়েকবার চোখ বোলালেই বাইনারি এবং দশমিকের পার্থক্য সহজেই চোখে পড়বে।

দশমিক	বাইনারি
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

দশমিক থেকে বাইনারিতে পরিবর্তন

এবার আমরা শিখবো কি ভাবে দশমিক সংখ্যাকে বাইনারি সংখ্যায় পরিবর্তিত করা যায়।

নিয়মটি হোল দশমিক সংখ্যাটিকে ক্রমাগত তুই দিয়ে ভাগ করে যেতে হবে। প্রতি ভাগের পর ভাগফল এবং একটি অবশিষ্ট থাকবে। প্রথম ভাগটি করার পর যে অবশিষ্ট থাকবে, সেটিকে এক জায়গায় লিখুন (অবশিষ্টের মান কেবল 1 আর 0 হতে পারে, অন্য কিছু কখনোই হবে না)। এবার ভাগফলকে পুনরায় 2 দিয়ে ভাগ করুন আর অবশিষ্টকে আগের অবশিষ্টের ঠিক বাঁ দিকে লিখুন। এই করতে করতে শেষ পর্যন্ত 0 কিম্বা 1 অবশিষ্ট থাকবে। সেটিকে একেবারে বাঁ দিকে লিখুন। ব্যাস, ওই অবশিষ্টগুলি সাজিয়ে, যে বাইনারি সংখ্যাটি সৃষ্টি হলো, সেটিই হবে ওই দশমিক সংখ্যার সমান মানের বাইনারি সংখ্যা।



বাইনারি সংখ্যা পদ্ধতি

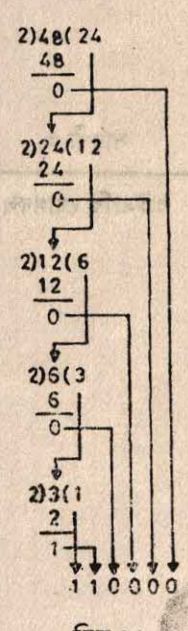
একটি উদাহরণে সমস্ত বিষয়টি পরিষ্কার হবে ঃ—

197 দশমিকের সমান বাইনারি সংখ্যা আমরা বার করবো। তাই প্রথমেই একে তুই দিয়ে ভাগ করতে হবে (চিত্র ১৫ দ্রষ্টব্য)।

সুতরাং 197 (দশমিক) = 11000101 (বাইনারি)।

আর একটা উদাহরণ দিলেই পুরো ব্যাপারটি পরিষ্কার হয়ে যাবে (চিত্র ১৬ দ্রষ্টব্য)—

সুতরাং 48 (দশমিক) = 110000 (বাইনারি)



চিত্র ১৬

দশমিক থেকে বাইনারিতে পরিবর্তন

বাইনারি যোগ

বাইনারি সংখ্যার যোগ বিয়োগ আমাদের পরিচিত দশমিকের যোগ বিয়োগের থেকে অনেক সহজ। দশমিক সংখ্যার যোগ বিয়োগ করতে হলে আমাদের মোট একশোটি যোগফল বা বিয়োগফল মুখস্ত রাখতে হয়। অবশ্য পুরো ব্যাপারটি আমাদের মনে এমন ভাবে গেঁথে গেছে যে মনেই হয় না কেবল মাত্র যোগ বা বিয়োগ করার জন্য আমাদের কম করে একশোটা যোগ বা বিয়োগফল মনে রাখতে হয়। বিশ্বাস না হয় সারণী 2কে দশ অবধি পূর্ণ করে দেখুন !

সারণী ২

1+0= 1	2+0= 2	3+0= 3	4+0= 4
1+1= 2	2+1= 3	3+1= 4	4+1= 5
1+2= 3	2+2= 4	3+2= 5	4+2= 6
1+3= 4	2+3= 5	3+3= 6	4+3= 7
1+4= 5	2+4= 6	3+4= 7	4+4= 8
1+5= 6	2+5= 7	3+5= 8	4+5= 9
1+6= 7	2+6= 8	3+6= 9	4+6=10
1+7= 8	2+7= 9	3+7=10	4+7=11
1+8= 9	2+8=10	3+8=11	4+8=12
1+9=10	2+9=11	3+9=12	4+9=13

দশমিক যোগের টেবিল

কিন্তু বাইনারি যোগ বিয়োগ করতে গেলে কেবল মাত্র চারটি যোগফল মনে রাখলেই চলে। সেই চারটি হলো—

সারণী ৩

0+ 0= 0
1+0= 1
0+1= 1
1+1=10

বাইনারি যোগফল

এই চারটি যোগফল মনে রাখলেই যে কোন বাইনারি যোগ বা বিয়োগ চোখ বুজে করে ফেলা যায়।

ধরা যাক আমরা বাইনারি 110101 এর সঙ্গে 100111কে যোগ করবো। তাহলে যোগফল হবে—

```
  110101
 +100111
 -------
 1011100
```

কি ভাবে হলো? প্রথমে ডান দিক থেকে যেমন আমরা দশমিক যোগ শুরু করতাম ঠিক সেই ভাবেই শুরু হল। 1+1=10 সুতরাং যোগফল 0 বসলো, হাতে রইলো 1। এবার দ্বিতীয় position-এ 0+1=1 হলো। তার সাথে হাতের 1-কে যোগ করলাম। যোগফল হলো 1+1=10। সুতরাং 0 বসলো যোগফলে, হাতে রইলো 1। এবার তৃতীয় position-এ 1+1=10 হলো। 10-এর 0-র সঙ্গে 1 যোগ করে হলো 11। তাই যোগফলের তৃতীয় position-এ 1 বসলো, হাতে রইলো 1। চতুর্থ position-এ ছিলো 0+0=0। তার সঙ্গে 1 যোগ করে হলো 1। সেটাই এবার যোগফলের চতুর্থ Position-এ বসলো। হাতে এবার কিছুই নেই। পঞ্চম position

এ আছে 1+0=1, তাই যোগফলের পঞ্চম position-এ বসালাম 1, হাতে এবার কিছু নেই। ষষ্ঠ position-এ ছিলো 1+1=10, সেটাই যোগফলের সপ্তম ও ষষ্ঠ position-এ বসলো।

এইভাবে দ্বিতীয় একটা যোগ করে ফেলা যাক।

```
  1010101
 +1100110
---------
 10111011
```

বাইনারি বিয়োগ

বিয়োগ করাও যোগের মতই সহজ। দুভাবে বিয়োগ করা যায়। আমরা শিখবো কমপ্লিমেণ্ট বার করে বিয়োগ করার কায়দা।

এভাবে বিয়োগ করতে গেলে পরপর তিনটে কাজ করতে হয়। ধরা যাক 1010101 থেকে 1100110 সংখ্যাকে আমাদের বিয়োগ করতে হবে। তাহলে প্রথমে

(১) 1100110-এর কমপ্লিমেনটারি (Complementary) বাইনারি সংখ্যাটি বার করতে হবে।

(২) 1010101-এর সঙ্গে 1100110-র কমপ্লিমেণ্টকে যোগ করতে হবে।

(৩) যোগফল যদি ওই দুটো সংখ্যার থেকে এক সংখ্যা (digit) বেশী হয় তবে ওই বাড়তি সংখ্যাকে সামনে না লিখে যোগফলের সঙ্গে যোগ করে দিতে হবে, আর সেটাই হবে উত্তর।

(৪) যদি যোগফলটি সংখ্যা দুটির থেকে বড় না হয়, তবে উত্তরের কমপ্লিমেণ্ট বার করে তার সামনে বিয়োগ চিহ্ন দিতে হবে।

কমপ্লিমেণ্ট

কমপ্লিমেণ্ট ব্যাপারটি খুব সোজা। কোনো বাইনারি সংখ্যার কমপ্লিমেণ্ট বার করতে হলে কেবল মাত্র সেই সংখ্যার 0-র জায়গায় 1 এবং 1 এর জায়গায় 0 লিখে দিলে যে সংখ্যাটি পাওয়া যাবে, সেটিই হবে কমপ্লিমেণ্ট। সুতরাং 1100110-র কমপ্লিমেণ্ট হবে 0011001।

আমরা এখন 1100110-র কপ্লিমেণ্ট কত হবে জেনে গেছি। সুতরাং এবার বিয়োগটি করে ফেলা যাক। 1010101 থেকে



বাইনারি সংখ্যা পদ্ধতি

1100110-কে বিয়োগ দিতে হলে আমাদের প্রথমেই 1010101-এর সঙ্গে 0011001-কে যোগ দিতে হবে।

$$\begin{array}{r} 1010101 \\ +0011001 \\ \hline 1101110 \end{array}$$

যোগফল 1010101-এর থেকে বড় নয়। সুতরাং ৪ নম্বর নিয়ম অনুযায়ী এই যোগফলের কমপ্লিমেণ্ট বার করে সামনে বিয়োগ চিহ্ন দিতে হবে।

তাহলে বিয়োগ ফল হবে $-0010001=-10001$

উত্তরটি হাতে নাতে মিলিয়ে নেওয়া যাক :—$(1010101)^2 =64+16+4+1=(85)_{10}$

$(100110)_2=64+32+4+2=(102)_{10}$

এবং$(-10001)_2=(-17)_{10}$

সুতরাং $85-102=-17$ উত্তর মিলেছে।

যদি 1100110-এর থেকে 1010101-কে বিয়োগ করা হয় তবে পদ্ধতিটা হবে :

$$\begin{array}{r} 1100110 \\ +0101010 \\ \hline 10010000 \end{array}$$

উত্তরটি 1100110-র থেকে বড় তাই যোগফলের সামনের 1-কে সামনে না লিখে বাকি যোগফলের সঙ্গে যোগ করতে হবে।

অর্থাৎ $10000+1=10001$

সুতরাং $1100110-1010101=10001$

উত্তর মিলিয়ে নিন, $(10001)_2=(17)_{10}$

অর্থাৎ $102-85=17$ উত্তর মিলেছে।

তাহলে যোগ বিয়োগ দেখা গেলো খুবই সোজা।

বাইনারি সংখ্যা এবং দশমিক সংখ্যা যেখানে একত্রে ব্যবহার করা হয়, সেখানে বাইনারি এবং দশমিকের পার্থক্য বোঝাতে Suffix 2 এবং 10 এর সাহায্য নেওয়া হয়। যেমন—$(101)_2$ বলতে বাইনারি 101 সংখ্যা এবং $(101)_{10}$ বলতে বোঝায় দশমিক 101 সংখ্যা।

বাইনারি গুণ এবং ভাগ

বাইনারি গুণ, ভাগ করাও দশমিক পদ্ধতির মত সোজা। দশমিক পদ্ধতিতে গুণ করতে হলে আমাদের 0 থেকে 9 অবধি নামতা মুখস্ত করতে হয়। কিন্তু বাইনারি গুণ ভাগ করার জন্য কেবল চারটে গুণফল মনে রাখলেই চলে। ৩৯ পৃষ্ঠায় বাইনারি সংখ্যার গুণের টেবিল দেওয়া হল।

```
0×0=0
0×1=0
1×0=0
1×1=1
```

তাহলে এবার একটি বড় দেখে গুণ করে ফেলা যাক ঃ

```
        1 1 0 0 1 1 0
         ×1 0 0 0 1
      _______________
        1 1 0 0 1 1 0
      0 0 0 0 0 0 0×
    0 0 0 0 0 0 0× ×
  0 0 0 0 0 0 0× × ×
1 1 0 0 1 1 0× × × ×
_____________________
1 1 0 1 1 0 0 0 1 1 0
```

আর একটি গুণ করা যাক ঃ—

```
        1· 0 1 0 1
       ×1 1 1  0 1
      ____________
        1·0 1 0 1
       0 0 0 0 0×
     1 0 1 0 1× ×
   1 0 1 0 1× × ×
 1 0 1 0 1× × × ×
__________________
1 0 0 1 ·1 0 0 0 0 1
```

তাহলে গুণ করাও খুবই সোজা তাই না ?

বাইনারি সংখ্যা পদ্ধতি

এবার বাকি রইলো ভাগ। এটার পদ্ধতিও ওই দশমিক পদ্ধতির ভাগের মতই একই রকম। তবে ভাগ করার জন্য মোটে হুটো ভাগফল মনে রাখতে হয়।

$0 \div 1 = 0$

এবং $1 \div 1 = 1$

(দশমিক পদ্ধতির মতই বাইনারিতে 0 দিয়ে 1 কিম্বা 0-কে ভাগ করা অবাস্তব হিসেবে ধরা হয়।)

তাহলে একটা ভাগ করে দেখা যাক—

বাইনারি ভাগ

```
111)100101(101· 01
    111
    ----
    1001
     111
    ----
     1000
      111
     ----
       10
```

হেক্সাডেসিম্যাল সংখ্যা পদ্ধতি

হেক্সাডেসিম্যাল সংখ্যা পদ্ধতি

কম্পিউটার বিজ্ঞানে বাইনারি সংখ্যা পদ্ধতি ছাড়াও আর একটি সংখ্যা পদ্ধতি খুব বেশী ব্যবহৃত হয়। এটি হলো হেক্সাডেসিম্যাল সংখ্যা পদ্ধতি (Hexadecimal Number System) বা সংক্ষেপে হেক্স (Hex)। হেক্সকে অবশ্য কম্পিউটারের হিসেব নিকেশের ভাষা হিসাবে ব্যবহার করা হয় না, এটিকে কেবল মাত্র কম্পিউটার প্রোগ্রামিং এর কাজেই ব্যবহার করা হয়।

হেক্সাডেসিম্যাল পদ্ধতির মূল সুবিধা হলো, এর র‍্যাডিক্স, বাইনারি সংখ্যা পদ্ধতির র‍্যাডিক্সের আট গুণ। ফলে বিরাট বিরাট বাইনারি সংখ্যাকে অনেক ছোট করে হেক্স সংখ্যায় লেখা যায়। সংখ্যা ছোট হয়ে যাবার দরুন প্রোগ্রামারের সেই সংখ্যাটি বলতে কিম্বা মনে রাখতেও অনেক সুবিধা হয়।

দ্বিতীয়ত হেক্সের র‍্যাডিক্স, বাইনারি সংখ্যা পদ্ধতির র‍্যাডিক্সের 4^{th} power এর সমান বলে হেক্স থেকে বাইনারি বা বাইনারি থেকে হেক্স সংখ্যা পদ্ধতিতে পরিবর্তন (Convertion), দশমিক পদ্ধতিতে পরিবর্তনের থেকে আরো বেশী সহজ। আসুন আমরা প্রথমে হেক্স সংখ্যা পদ্ধতির সঙ্গে পরিচিত হই।

আমরা আগেই জেনেছি, হেক্স পদ্ধতির র‍্যাডিক্স, বাইনারি সংখ্যা পদ্ধতির র‍্যাডিক্সের 8 গুণ: অর্থাৎ $2 \times 8 = 16$ সুতরাং এই পদ্ধতিতে মোট ষোলটি সংখ্যা আছে যাদের বিভিন্ন ভাবে সাজিয়ে আমরা হেক্স পদ্ধতির সমস্ত সংখ্যা লিখতে পারি। এই ষোলটি সংখ্যা হলো 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, এবং F. কি হলো ? সব গুলিয়ে গেল না ? 1, 2, 3, 4 তো ঠিক আছে, কিন্তু তার সঙ্গে আবার A, B, C, D লেখা হলো কেন ?

আসলে হেক্স পদ্ধতিতে A, B, C, D, E এবং F হলো এক একটি সংখ্যা যাদের মান দশমিক পদ্ধতিতে যথাক্রমে 10, 11, 12, 13, 14, 15-র সমান। তাই হেক্স পদ্ধতিতে F লিখলে সেটি দশমিক পদ্ধতির 15-র সম মানের সংখ্যা বোঝায়। পূর্বের পাতায় দশমিক, হেক্স এবং বাইনারি সংখ্যা পদ্ধতিতে দশমিক পদ্ধতির 0 থেকে 15 মানের সংখ্যার একটি তালিকা দেওয়া হলো। এর থেকে এক নজরেই তিনটি সংখ্যা পদ্ধতির পারস্পরিক সম্পর্কটি বোঝা যাবে।

দশমিক	হেক্স	বাইনারি		
0	0	0	অথবা	0000
1		1	"	0001
2	2	10	"	0010
3	3	11	"	0011
4	4	100	"	0100
5	5	101	"	0101
6	6	110	"	0110
7	7	111	"	0111
8	8	1000		
9	9	1001		
10	A	1010		
11	B	1011		
12	C	1100		
13	D	1101		
14	E	1110		
15	F	1111		



হেক্সাডেসিম্যাল সংখ্যা পদ্ধতি

সারণী ৪

দশমিক, হেক্স এবং বাইনারি সম্পর্ক

দশমিক সংখ্যা পদ্ধতিতে যেমন 0 লেখার পরই 10 আসে। ঠিক তেমনি হেক্স পদ্ধতিতে F সংখ্যার পরই আসে 10 (অবশ্য হেক্স 10 এর মান দশমিক 16-র সমান)। তারপর 11, 12, 13,···1C, 1D, 1E, 1F হয়ে হয় 20। 20-র পর 21, 22, 23,···2C, 2D, 2E, 2F হয়ে হয় 30। এই ভাবে হেক্স সংখ্যা পদ্ধতি এগিয়ে চলে। নিচে দশমিক পদ্ধতির 0 থেকে 256 পর্যন্ত মানের সংখ্যার সমান হেক্স এবং বাইনারি সংখ্যার চেহারাটি কেমন হয় সেটি দেখানো হলো।

সারণী ৫

দশমিক 0 থেকে 256-র সম মানের হেক্স এবং বাইনারি সংখ্যা

দশমিক	হেক্স	বাইনারি
0	0	0
1	1	1
2	2	10
⋮	⋮	
9	9	1001
10	A	1010
⋮	⋮	⋮
15	F	1111

হেক্সাডেসিম্যাল সংখ্যা পদ্ধতি

16	10	10000
17	11	10001
⋮	⋮	⋮
25	19	11001
26	1A	11010
27	1B	11011
⋮	⋮	⋮
31	1F	11111
32	20	100000
33	21	100001
⋮	⋮	⋮
159	9F	10011111
160	AO	10100000
161	A1	10100001
⋮	⋮	⋮
254	FE	11111110
255	FF	11111111
256	100	100000000

সারণী ৫

দশমিক, হেক্স এবং বাইনারির পারস্পরিক সম্পর্ক।

এই তালিকা থেকে স্পষ্টই দেখা যাচ্ছে হেক্স সংখ্যার র্যাডিক্স বড় বলে অনেক বড় দশমিক সংখ্যাও কত ছোট আর সহজ ভাবে হেক্স পদ্ধতিতে লেখা সম্ভব হয়েছে। আর যদি ওই একই সংখ্যা আমাদের বাইনারি সংখ্যায় লিখতে হত, তবে তালিকা থেকে দেখতেই পাচ্ছেন, কি বিরাট একটা বাইনারি সংখ্যার সৃষ্টি হত। এতো বড়ো একটি সংখ্যা নির্ভুল ভাবে লেখা এবং মনে রাখা, সবার পক্ষেই কষ্টকর। 8 বিট পারশোনাল মাইক্রো কম্পিউটারের কথাই ধরুন না, এই ধরনের সমস্ত কম্পিউটারে যাবতীয় তথ্য এবং নির্দেশ 8 বিট বাইনারি সংখ্যার মাধ্যমে আদান প্রদান করা হয়। তাই এ ধরনের কম্পিউটারকে মেশিন ল্যাংগোয়েজে প্রোগ্রাম করার দরকার পড়লে (অনেক সময় খুব বড় এবং জটিল প্রোগ্রাম যেমন স্পেস ইনভেডার ধরনের ভিডিও গেম কিম্বা দ্রুত কম্পিউটার গ্রাফিক্স সৃষ্টি করতে প্রায়ই আমাদের হাই লেভেল ল্যাংগোয়েজের বদলে মেশিন

ল্যাংগোয়েজ ব্যবহার করতে হয়) কম্পিউটারের সমস্ত নির্দেশ এবং তথ্য আমাদের 8 বিট বাইনারি সংখ্যায় লিখতে হয় এবং মনে রাখতে হয়। অত্যাশ্চর্য স্মৃতি শক্তি এবং একাগ্রতা না থাকলে এই ভাবে প্রোগ্রাম করা যে কোনো প্রোগ্রামারের কাছে প্রায় অসম্ভব কাজই বলা চলে। উদাহরণ স্বরূপ একটি 8 বিট্ বাইনারি সংখ্যা 11010010-র কথাই ধরুন না, এরকম শয়ে শয়ে বিভিন্ন বাইনারি সংখ্যা পর পর নির্ভুল ভাবে লিখতে গেলে ভুল হবার সম্ভাবনাই বেশী। অথচ এই বাইনারি সংখ্যাটিকেই যদি হেক্সে রূপান্তরিত করা হয় তবে যে সংখ্যাটি পাওয়া যায় সেটি হলো D 2। তাহলেই দেখুন 11010010-র বদলে যদি কেবল মাত্র কী বোর্ডের D এবং 2 এই দুটি বোতাম টিপলেই কাজ চলে তবে প্রোগ্রামারের কাজ কতো সহজ হয়ে যায়।

এতো গেল কেবল মাত্র 8 বিট্ মাইক্রো কম্পিউটারের কথা। আর যদি আরো বড় 16 বিট্ মিনি কম্পিউটার কিম্বা 32 বিট্ মেইনফ্রেম কম্পিউটারের কথা ভাবা যায় তবে বুঝতেই পারছেন মেশিন ল্যাংগোয়েজে প্রোগ্রাম করতে হেক্স সংখ্যা পদ্ধতির ব্যবহার কি ভীষণ প্রয়োজনীয়।

এখন প্রশ্ন উঠতে পারে আমরা তো বাইনারি সংখ্যার সম মানের দশমিক সংখ্যাও ব্যবহার করে প্রোগ্রাম করতে পারতাম। সেই সংখ্যাটিও বাইনারির থেকে অনেক ছোট হতো। তবে হেক্সকে বেছে নেওয়ার কারণ কি? এর উত্তর বাইনারি থেকে দশমিক বা দশমিক থেকে বাইনারিতে পরিবর্তন করা যত সময়সাপেক্ষ, হেক্স থেকে বাইনারি বা বাইনারি থেকে হেক্সে পরিবর্তন তার তুলনায় জলের মতো সোজা। ৪১ নম্বর পাতায় দেওয়া সারণী 8 সামনে থাকলে এই পরিবর্তন মুখে মুখে করা যায়। প্রথমে দেখা যাক কি ভাবে হেক্স থেকে বাইনারিতে পরিবর্তন করা হয়।

হেক্স থেকে বাইনারিতে পরিবর্তন

যে কোনো একটি হেক্স সংখ্যা চিন্তা করা যাক। ধরুন 5E হলো সেই সংখ্যাটি। এরই সমান বাইনারি সংখ্যা বার করতে

হবে। এর জন্য প্রথমেই ৪১ পাতার সারণী 8 থেকে 5 হেক্সের সমান বাইনারি সংখ্যাটি বার করে কাগজে লিখে ফেলুন। এটি হবে 0101। তার পর টেবিল থেকে E হেক্স সংখ্যার সমান বাইনারি সংখ্যাটি দেখে 0101 এর ডান দিকে লিখুন। এটি হবে 1110। ব্যাস, এবার মোট যে বাইনারি সংখ্যাটি পেলেন সেটিই হলো 5E হেক্সের সমান বাইনারি সংখ্যা।

অর্থাৎ 5E = 01011110

একই ভাবে বার করে ফেলুন 9FF হেক্সের সমান বাইনারি সংখ্যাটি, দেখবেন উত্তরটি হবে—

9FF = 100111111111

বাইনারি থেকে হেক্সে পরিবর্তন

এবার দেখা যাক বাইনারি থেকে কি ভাবে হেক্স সংখ্যা বার করা যায়। এরজন্য প্রথমেই যে বাইনারি সংখ্যাটির হেক্স বার করতে হবে তাকে ডান দিক থেকে 4 সংখ্যা (Bit) করে আলাদা করে নিতে হবে। যেমন আমরা যদি 101010111 বাইনারি সংখ্যাটির হেক্স বার করতে চাই তবে প্রথমেই এই সংখ্যাটিকে 4 বিট্ করে তিন ভাগে ভাগ করতে হবে।

101010111 = 1/0101/0111

এবার ওই একই টেবিল থেকে 1, 1010 এবং 0111-এর সমান হেক্স সংখ্যাগুলি খুঁজে যথাক্রমে সাজালেই উত্তর বেরিয়ে যাবে। দেখুন, টেবিল থেকে দেখা যাচ্ছে—

1 = 1 (হেক্স)
0101 = 5 (হেক্স)
0111 = 7 (হেক্স)

সুতরাং বাইনারি 101010111 = 157 (হেক্স)

আর একটি উদাহরণ দেখা যাক। বাইনারি 11011110 সংখ্যার হেক্স আমাদের বার করতে হবে। প্রথমেই একে 4 বিট্ করে ভাগ করা যাক—

11011110 = 1101/1110

টেবিলে দেখা যাচ্ছে 1101 = D
এবং 1110 = E

সুতরাং বাইনারি 11011110 = হেক্স D E

পরিচ্ছেদ ৪

ইনপুট ডিভাইস

আমরা আগের অধ্যায়ে জেনে গেছি, কম্পিউটার বাইনারি সংখ্যা পদ্ধতিতে কাজ করে। তাই কম্পিউটারের স্মৃতির ভাণ্ডারে কোনো তথ্য পাঠাতে হলে সেই তথ্যকে প্রথমে বাইনারি কোডে পরিবর্তিত করে ডিজিটাল বিদ্যুৎ সিগন্যালের মাধ্যমে পাঠাতে হয়। যে যন্ত্র এই কাজটি করে থাকে তাকে বলে ইনপুট ডিভাইস।

আমরা ইতিমধ্যে বেশ কয়েকটি ইনপুট ডিভাইসের নামও জেনে গেছি যেমন, কী বোর্ড, পাঞ্চ কার্ড, পাঞ্চ টেপ ইত্যাদি। এছাড়া আছে ম্যাগনেটিক ইঙ্ক রিডার, অপটিকাল মার্ক রিডার, অপটিকাল ক্যারেকটার রেকগনাইজার, ম্যাগনেটিক টেপ এনকোডার, এ ডি কনভার্টার প্রভৃতি। এরা প্রত্যেকেই বিভিন্ন ধরণের প্রয়োজনীয় তথ্যকে ডিজিটাল সিগন্যালে পরিবর্তিত করে কম্পিউটারের স্মৃতি ভাণ্ডারে পাঠিয়ে দেয়। আসুন, এই সমস্ত ইনপুট ডিভাইস কি করে কাজ করে জানা যাক।

কী বোর্ড

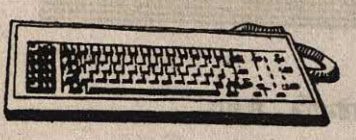
কম্পিউটারের সব থেকে জনপ্রিয় ইনপুট ডিভাইস হলো কী বোর্ড। ছোট বড় সমস্ত কম্পিউটারেই অন্তত একটি কী বোর্ড ইনপুটের ব্যবস্থা সব সময়ই রাখা হয়। বড় কম্পিউটারের ক্ষেত্রে অনেক সময় এই ইনপুট ডিভাইসকে কনসোলও (Console) বলা হয়।



কী বোর্ড

কম্পিউটার কী বোর্ড বা কনসোলের সঙ্গে টাইপরাইটারের আকৃতি কত সাদৃশ্য আছে। তবে এর চাবিগুলি সবই বিশেষ ধরনের ইলেকট্রিক সুইচ। প্রতিটা সুইচের ওপর এক বা একাধিক ক্যারেকটার এবং হাই লেভেল কম্যাণ্ড বা স্টেটমেণ্ট লেখা থাকে। ওই সুইচ বা চাবি টিপে কম্পিউটারে নির্দেশ বা প্রোগ্রাম পাঠানো হয়।

ধরা যাক, যদি কোনো অ্যাসকি কী বোর্ডে A লেখা বোতামটি টেপা হয়, অমনি অ্যাসকি কী বোর্ড থেকে এক সারি বৈদ্যুতিক স্পন্দন বেরিয়ে আসে, যাকে কাগজে বাইনারি সংখ্যায় লিখলে হয়, 1000001 (এই বাইনারি সংখ্যাটি অ্যাসকি কোড অনুসারে ইংরিজি A হরফকে নির্দেশ করে)। তাই আমরা যখনই কোন কম্পিউটারের অ্যাসকি কী প্যাডে কোন বোতাম টিপি, সঙ্গে সঙ্গে কী প্যাডের সঙ্গে সংযুক্ত এনকোডার (Encoder) নামে একটি ইলেকট্রনিক সার্কিট সেই বোতামের ওপর লেখা সংখ্যা, হরফ কিম্বা নির্দেশকে অ্যাসকি কোড অনুযায়ী 5 ভোল্ট এবং 2 ভোল্টের এক সারি বৈদ্যুতিক স্পন্দনে (Digital Pulse Train) পরিণত করে।



চিত্র ১৭

অ্যাসকি কী বোর্ড

সাধারণত সমস্ত কম্পিউটারের ইনপুট ডিভাইস থেকে আসা বিদ্যুৎ স্পন্দনের মধ্যে 5 ভোল্ট বিভবকে বাইনারি 1 ধরা হয়, এবং ·2 ভোল্ট বিভবকে বাইনারি 0 ধরা হয়। কম্পিউটার এই বিদ্যুৎ স্পন্দনের সারি থেকে বুঝে যায় তাকে কি নির্দেশ বা তথ্য পাঠানো হচ্ছে।

এছাড়া কী বোর্ডের চাবির সঙ্গে লাগানো থাকে কনট্যাক্ট ডিবাউন্সার সার্কিট। কনট্যাক্ট ডিবাউন্সারের কাজ হলো একবার চাবি টেপা হলে যাতে একবারই আউটপুট পাওয়া যায় তাকে স্থির নিশ্চিত করা।

বর্তমানে আমাদের স্কুলে বা কলেজে যে সব কম্পিউটার আমাদের ব্যবহার করতে হচ্ছে বা হবে তাকেও আমাদের কী বোর্ডের মাধ্যমেই প্রোগ্রাম করতে হয়।

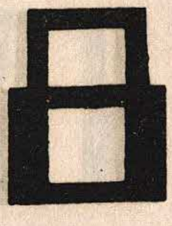
অ্যাসকি কোড

অ্যাসকি কোডের সারণী ৩১ পৃষ্ঠায় দেওয়া আছে। ওই সারণীতেই দেখানো আছে কি ভাবে বিভিন্ন হরফ, ক্যারেকটার এবং কম্যাণ্ডের অ্যাসকি কোড খুঁজে বার করতে হয়।

পাঞ্চ টেপ

পাঞ্চ টেপ হলো মেইন ফ্রেম কম্পিউটারের একটি অন্যতম প্রধান ইনপুট ডিভাইস। এই ব্যবস্থায় আধ ইঞ্চি থেকে তিন ইঞ্চি চওড়া শক্ত কাগজ কিম্বা প্লাস্টিকের লম্বা ফিতেতে ফুটো করে কম্পিউটারের প্রোগ্রাম

লিপিবদ্ধ করা হয়। ফিতের ওপর আটটি কিম্বা ষোলোটি চ্যানেল বরাবর ফুটো করা হয়। প্রতি লাইনে এই আটটি বা ষোলটি চ্যানেলের কোনো কোনোটাতে ফুটো করে এবং কোনোটাতে ফুটো না করে বিভিন্ন অক্ষর, সংখ্যা, স্পেশ্যাল ক্যারেকটার কিম্বা নির্দেশ বোঝানো হয়। এটার ব্যাপারেও অ্যাসকি কোডের মতো আন্তর্জাতিক কোড মেনে চলা হয়। এ ধরনের টেপের ব্যবহার টেলেক্স মেশিনেও দেখা যায়।



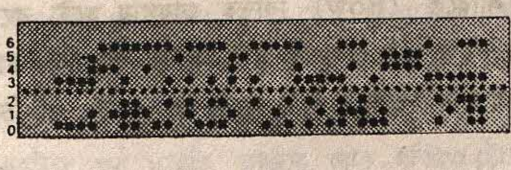
পাঞ্চ টেপ

পাঞ্চ টেপে ফুটো করার জন্য কী বোর্ড নামে একটি টাইপ রাইটারের মতো যন্ত্র ব্যবহার করা হয়। এতে টাইপ করলে যন্ত্রের ভেতর রাখা অক্ষত ফিতেতে, সংখ্যা অক্ষর কিম্বা নির্দেশ অনুসারে ঠিক ঠিক জায়গায় ফুটো হয়ে যায়। কী বোর্ড যন্ত্রে টাইপ রাইটারের মতো লেখা ছাপাবার ব্যবস্থাও থাকে। ফলে কী বোর্ডে যে যে বোতাম টেপা হল, সেগুলি একটা সাদা কাগজে ছাপা হয়ে বেরিয়ে আসে। পুরো প্রোগ্রাম টাইপ করার পর ওই সাদা কাগজে ছাপা লেখা অর্থাৎ হার্ড কপি (Hard Copy) কে মূল প্রোগ্রামের সঙ্গে মিলিয়ে দেখা হয়। ভুল থাকলে আবার করে টেপ পাঞ্চ করা হয়।

পাঞ্চ টেপে সঠিক সংকেত ফুটো করা হয়ে গেলে তাকে পাঞ্চ টেপ রিডার যন্ত্রে পাঠানো হয়। এই যন্ত্রের ইলেকট্রনিক বা মেকানিকাল ব্যবস্থা, পাঞ্চ টেপের প্রতি লাইনের কোন্ কোন্ চ্যানেলে ফুটো আছে বা নেই তা বুঝে নেয় এবং তার ভেতরের এনকোডার সার্কিট প্রতি ফুটোর জায়গায় বাইনারি 1 এবং ফুটো না থাকা জায়গায় বাইনারি 0 হিসেবে ডিজিটাল সিগন্যাল তৈরী করে কম্পিউটারের আভ্যন্তরীণ স্মৃতি ভাণ্ডারে পাঠিয়ে দেয়। হারমান হলেরিথই সর্বপ্রথম এই পদ্ধতিতে কম্পিউটার প্রোগ্রামিং করার ব্যবস্থা করেছিলেন।

আধুনিক মেকানিকাল টেপ রিডার যন্ত্রে ফুটো বোঝবার জন্য

চিত্র ১৮

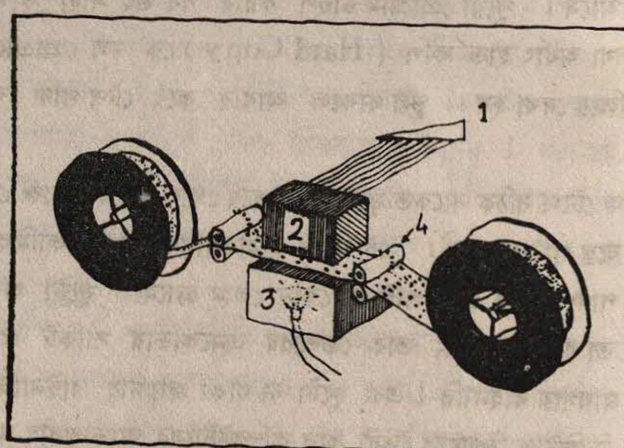


পাঞ্চ টেপ



পাঞ্চ কার্ড

সেনসিং পিন্ (Sensing Pin) এবং স্টার হুইল ব্যবহার করা হয়। দ্রুতগতি সম্পন্ন ইলেকট্রিক টেপ রিডারে ব্যবহার করা হয় ফোটো ইলেকট্রিক সেল বা ফোটো ডায়োড। এই ব্যবস্থায় টেপের একদিকে রাখা হয় একটি উজ্জ্বল ইলেকট্রিক আলো আর অন্য দিকে প্রতি চ্যানেলের জন্য থাকে একটি করে ফোটো ইলেকট্রিক সেল বা ফোটো ডায়োড। টেপে ফুটো থাকলে বাল্বের আলো ফুটোর মধ্যে দিয়ে টেপের উল্টো দিকে রাখা ফোটো সেল বা ডায়োডে পড়ে। আলো পড়লেই সেল বা ডায়োডের মধ্যে দিয়ে বিদ্যুৎ প্রবাহ বেড়ে যায় আর বাইনারি 1 ডিজিটাল সংকেত পাওয়া যায়। ফুটো না থাকলে আলো পড়ে না আর সেই চ্যানেলের সেল বা ডায়োড থেকে বাইনারি 0 সংকেত পাওয়া যায়। এই ব্যবস্থায় টেপের সঙ্গে সংকেত পড়া ব্যবস্থার সরাসরি কোনো যান্ত্রিক সংযোগ না থাকায় খুব দ্রুত এবং নির্ভুলভাবে টেপের তথ্য কম্পিউটারে পাঠানো যায়।



চিত্র ১৯

ফোটোইলেক্ট্রিক পাঞ্চ টেপ রিডার

1—কম্পিউটারে তথ্য পাঠানোর তার গুচ্ছ (**Data Bus**)
2— ফোটো সেল
3—আলোর উৎস
4—স্প্রকেট ড্রাইভ

পাঞ্চ কার্ড

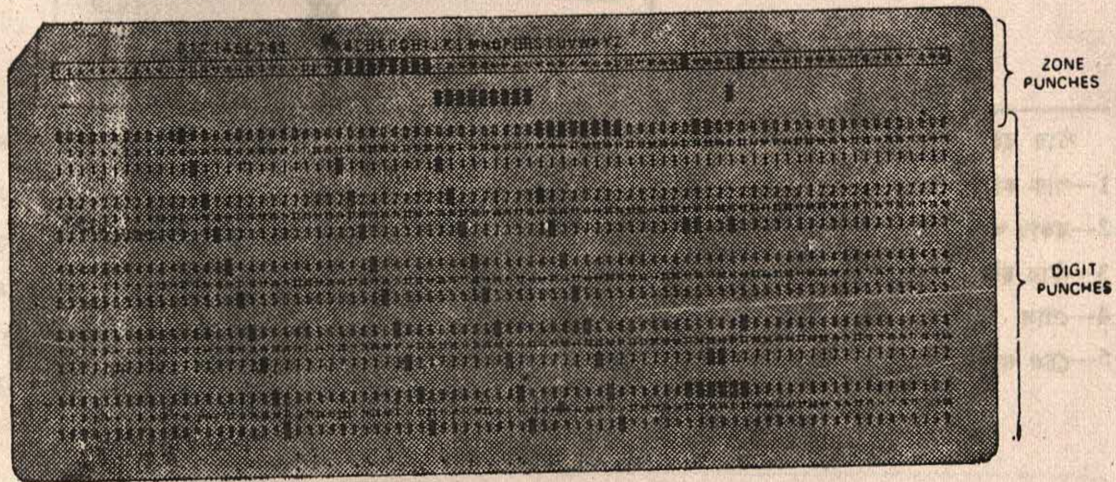
বড় কম্পিউটারের ইনপুট ডিভাইস হিসেবে পাঞ্চ কার্ডের ব্যবহার প্রায় সর্বাধিক। 8 সেমি চওড়া আর 19 সেমি লম্বা পাতলা অথচ শক্ত কাগজে তৈরী হয় এই কার্ড। প্রতি কার্ডে 80 টা স্তম্ভ (Column) এবং 12টা সারিতে (Row) বিভিন্ন জায়গায় ফুটো করে বিভিন্ন ইংরিজি হরফ, সংখ্যা এবং স্পেশাল ক্যারেকটার প্রোগ্রাম করা হয়।

কার্ডে প্রোগ্রাম অনুসারে ফুটো করার জন্য কার্ড পানচিং মেশিন

ব্যবহার করা হয়। কার্ড পাঞ্চের সঙ্গে সঙ্গে মেশিনে সমস্ত প্রোগ্রামের হার্ড কপিও ছাপা হয়ে যায়। হার্ড কপির সঙ্গে মূল প্রোগ্রাম মিলিয়ে দেখে নেওয়া হয় সব ঠিক ঠিক টাইপ করা হয়েছে কিনা। প্রোগ্রাম মিলিয়ে (Verify) দেখার জন্য অনেক জায়গায় পাঞ্চ কার্ডকে ভেরিফায়ার মেশিনে চালানো হয়। এই যন্ত্র, কার্ডের ফুটো অনুসারে লিপিবদ্ধ প্রোগ্রামকে ইংরিজি হরফ সংখ্যা এবং স্পেশাল ক্যারেকটারে ছাপিয়ে হার্ড কপি তৈরী করে দেয়।



পাঞ্চ কার্ড



চিত্র ২০

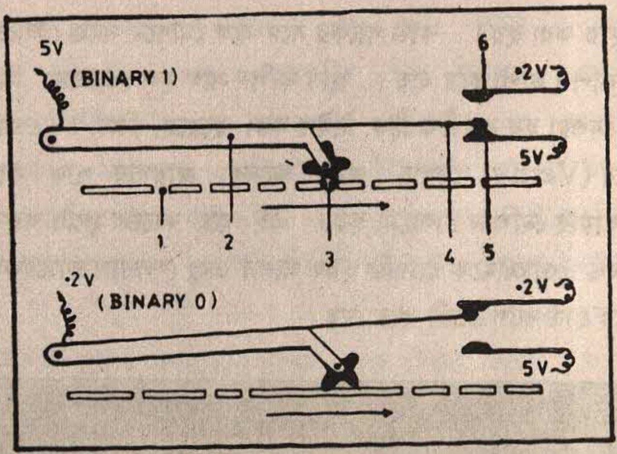
পাঞ্চ কার্ড

পাঞ্চ টেপের সঙ্গে পাঞ্চ কার্ডের প্রধান পার্থক্য হলো পাঞ্চ টেপে একটি সম্পূর্ণ প্রোগ্রাম পরপর পাঞ্চ করা যায়, কারণ টেপ যতখুশি লম্বা হতে পারে। কিন্তু পাঞ্চ কার্ডে 80 টা কলম থাকার দরুণ একটা কার্ডে সর্বাধিক 80 টা সংখ্যা, হরফ বা স্পেশাল ক্যারেকটার লেখা যেতে পারে, ফলে একটি সম্পূর্ণ প্রোগ্রাম পাঞ্চ করতে একাধিক কার্ড লেগে যায়। এক একটা কার্ডে প্রোগ্রামের কেবল একটা লাইন লেখা হয়। সুতরাং যন্ত্রে ঢোকাবার আগে পাঞ্চ কার্ডকে পরপর সঠিক ক্রমে সাজিয়ে রাখতে হয়।

পাঞ্চ কার্ড রিডারের কার্যপ্রনালী পাঞ্চ টেপ রিডারেরই অনুরুপ। কার্ডগুলিকে কার্ড রিডার যন্ত্রের হপারে রাখা হয়। হপারের নিচেই থাকে রবারের রোলার। কার্ড রিডার অন্ করলেই ওই রোলারগুলো হপার থেকে একটার পর একটা কার্ডকে দ্রুত গতিতে টেনে নিয়ে ফুটো সংকেত পড়ার ইলেকট্রোমেকানিক্যাল কিম্বা ফটো ইলেকট্রিক



পাঞ্চ কার্ড



চিত্র ২১

স্টার হুইল কার্ড রিডিং ব্যবস্থা।

1—পাঞ্চ কার্ডের ফুটো

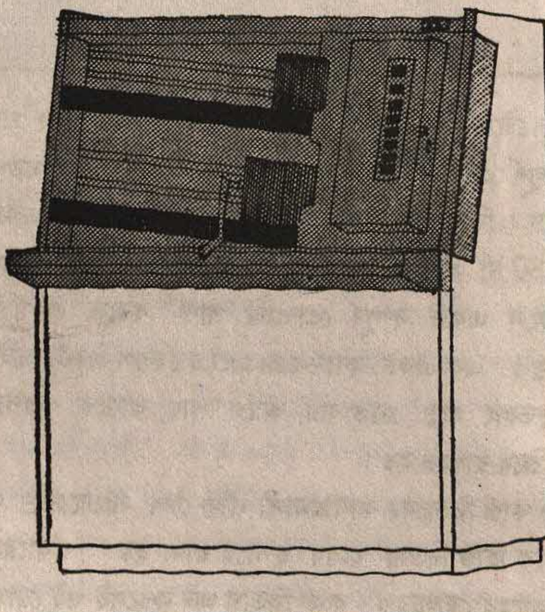
2—ব্রকার আর্ম

3—স্টার হুইল

4—পোল

5—চেঞ্জ ওভার কণ্ট্যাক্ট

ব্যবস্থার মধ্যে দিয়ে টেনে নিয়ে যায়। এক একটা কার্ড পড়া হয়ে গেলে সেগুলো রোলারের মাধ্যমে আউটপুট হপারে ছিটকে বেরিয়ে আসে এবং পরপর আবার স্তুপাকৃত ভাবে জমা হতে থাকে। আধুনিক কার্ড রিডার গুলি প্রতি সেকেণ্ডে 12টি থেকে, শুরু করে 34টি কার্ড অনায়াসে পড়ে ফেলে। পুরো ঘটনাটা এত দ্রুত ঘটতে থাকে যে রোলারের মধ্যে দিয়ে ছুটে যাওয়া কার্ডগুলো আলাদা করে বোঝাই যায় না, মনে হয় পাঞ্চ কার্ডের একটা স্রোত ছুটে চলেছে!



চিত্র ২২

পাঞ্চ কার্ড রিডার যন্ত্র

এম. আই. সি. আর (M. I. C. R)

কম্পিউটার যাতে সরাসরি বিশেষ কালিতে ছাপানো হরফ বা সংখ্যা পড়তে পারে, তারই ব্যবস্থা হলো ম্যাগনেটিক ইংক ক্যারেকটার রেকগনাইজার ইনপুট ডিভাইস (Magnetic Ink Character Recognizer) বা সংক্ষেপে MICR ইনপুট ডিভাইস।



এম. আই. সি. আর

এই ব্যবস্থায় সাধারণ কাগজের ওপর ফেরো ম্যাগনেটিক বস্তুর গুঁড়ো দিয়ে তৈরী কালো কালিতে সমস্ত হরফ এবং সংখ্যা ছাপানো হয়।



চিত্র ২৩

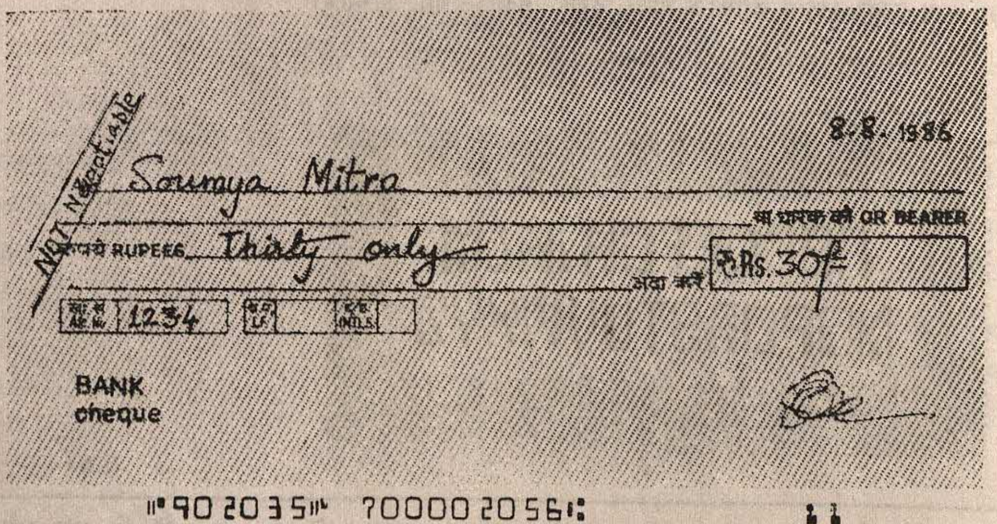
এম. আই. সি. আর ক্যারেকটার

এম. আই. সি. আর যন্ত্রের মধ্যে দিয়ে বিশেষ কালিতে লেখা এই কাগজকে চালালে প্রথমে যন্ত্রের অভ্যন্তরের একটি শক্তিশালী চুম্বক সমস্ত ছাপানো ক্যারেকটারকে চুম্বকে পরিণত করে। চুম্বকত্ব প্রাপ্ত ক্যারেকটারগুলিকে তারপর টেপ রেকর্ডারের হেডের মত একটা বিশেষ হেডের তলা দিয়ে চালান হয়। তড়িৎ চুম্বকীয় আবেশের ফলে হেডে ক্ষীণ আবিষ্ট তড়িৎ সৃষ্টি হয়। অ্যামেরিকান ব্যাংকিং অ্যাসোশিয়েশন এম. আই. সি. আর ক্যারেকটারগুলি এমন বিশেষভাবে ডিজাইন করেছেন যাতে প্রতিটা আলাদা ক্যারেকটার হেডের মধ্যে বিভিন্ন নির্দিষ্ট ভোল্টেজের তড়িৎ আবিষ্ট করে। অর্থাৎ A হরফ হেডের তলা দিয়ে যাবার সময় যে রকম তড়িৎ আবিষ্ট করে তার সঙ্গে B হরফের আবিষ্ট তড়িতের পার্থক্য থাকে। রিডারের অভ্যন্তরের এনকোডার সার্কিট এই পার্থক্য থেকে বুঝে নেয় কাগজে কোন্ ক্যারেকটারটি লেখা আছে এবং সেই অনুসারে ডিজিটাল সিগন্যাল তৈরী করে কম্পিউটারে পাঠায়।



ও. সি. আর

আমাদের দেশের ইদানীং এর ব্যবহার বেশ চালু হয়েছে। কয়েকটি ব্যাঙ্কের চেকে এবং ক্রেডিট কার্ডে অ্যামেরিকান ব্যাংকিং অ্যাসোশিয়েশন কর্তৃক নির্দিষ্ট MICR হরফে অ্যাকাউন্ট নাম্বার, ব্রাঞ্চ নাম্বার ইত্যাদি ছাপা হচ্ছে। এই ব্যবস্থার ফলে ওই সব সংস্থা তাদের হিসেব নিকেশের কাজ স্বয়ংক্রিয় ভাবে কম্পিউটারের সাহায্যে করতে পারছেন।



চিত্র ২৪

এম. আই. সি. আর
যুক্ত ব্যাঙ্ক চেক্

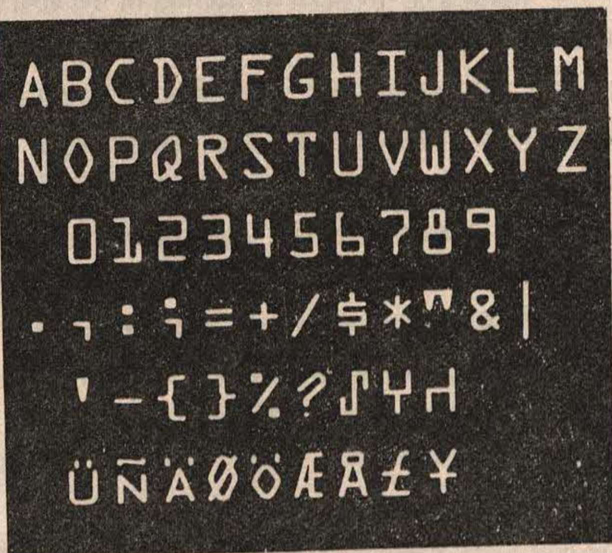
ও. সি. আর (O. C. R)

সাধারণ কাগজে সাধারণ কালিতে ছাপা অক্ষর যাতে কম্পিউটার পড়তে পারে তারই ব্যবস্থা হলো অপটিকাল ক্যারেকটার রেকগনাইজার বা সংক্ষেপে O.C.R. এই ব্যবস্থাকেই বর্তমানে কম্পিউটার প্রযুক্তিবিদ্‌রা আরো বহুগুণ উন্নত করার চেষ্টা করছেন। অপটিকাল ক্যারেকটার রেকগনাইজার যন্ত্রে পড়বার জন্য যে তথ্য বা সংখ্যা ছাপানো হয়, সেটাও MICR এর মতোই একটা বিশেষ বিদঘুটে টাইপে ছাপতে হয়। যে কোনো রকম টাইপ ফেসে ছাপা তথ্য অপটিকাল ক্যারেকটার রেকগনাইজার পড়তে পারে না। তবে ইদানিং অ্যামেরিকান পোস্ট অফিসে এক উন্নত ধরনের অপটিকাল ক্যারেকটার রেকগনাইজার ব্যবহার করা হচ্ছে যেটি হাতে লেখা অক্ষরও পড়তে পারে।

অপটিকাল ক্যারেকটার রেকগনাইজারে থাকে উজ্জ্বল আলো, লেন্স আর একটি আয়তক্ষেত্রাকার ফোটো ডায়োডের ম্যাট্রিক্স। কাগজে ছাপানো অক্ষরের উপর আলো ফেলে লেন্সের সাহায্যে তার প্রতিবিম্ব তৈরী করে ওই ডায়োড ম্যাট্রিক্সের ওপর ফেলা হয়। ডায়োড ম্যাট্রিক্সের ওপর আলো পড়লেই তার থেকে বৈদ্যুতিক সংকেত পাওয়া যায়। ওই সংকেতের সঙ্গে অপটিকাল ক্যারেকটার রেকগনাইজারের ছোট্ট স্মৃতি ভাণ্ডারে সঞ্চিত বিভিন্ন অক্ষর এবং সংখ্যার সংকেত মিলিয়ে দেখা হয়। যে সংখ্যার সঙ্গে সেই সংকেত মেলে, রেকগনাইজারের এনকোডার সেই হরফ বা সংখ্যার ডিজিটাল সংকেত কম্পিউটারে পাঠায়।



A/D কনভার্টার



চিত্র ২৫

ও. সি. আরক্যারেকটার

A/D কনভার্টার

A/D কথাটির উৎপত্তি Analog to Digital থেকে। A/D কনভার্টার হলো এমন একটি ইলেক্ট্রনিক সার্কিট যেটি অ্যানালগ ভোল্টেজকে মেপে তার মান ডিজিটাল সংকেতে জানিয়ে দেয়।

কম্পিউটারের জন্য তাই A/D কনভার্টার হলো একটি দারুন প্রয়োজনীয় ইনপুট ব্যবস্থা। এর সাহায্যে কম্পিউটার বিভিন্ন পরিমাপক ট্রান্সডিউসার (Transducer) থেকে আগত অ্যানালগ সিগন্যালকে সরাসরি বুঝতে পারে। যেমন তাপমাত্রা মাপার ট্রান্সডিউসার



বার কোড

(থার্মোকাপল্, থার্মিস্টর ইত্যাদি) কে যদি একটি A/D কনভার্টারের মাধ্যমে কম্পিউটারের সঙ্গে যুক্ত করা যায় তবে বাইরের তাপমাত্রার কখন কেমন পরিবর্তন হচ্ছে সেটা কম্পিউটার স্মৃতিতে জমা হয়ে যায়। আমরা তখন কম্পিউটারকে বিশেষ প্রোগ্রাম করে এই তথ্য অনুযায়ী বিভিন্ন কাজ করিতে নিতে পারি (যেমন বয়লারের তাপমাত্রা নিয়ন্ত্রণ; ফার্ণেসের জ্বালানী জোগান ইত্যাদি)। এতো গেলো একটা উদাহরণ। বস্তুত পক্ষে A/D কনভার্টার ইনপুট ব্যবস্থার যে কত বিচিত্র প্রয়োগ আছে, তা বলে শেষ করা যায় না। বিভিন্ন কল কারখানায়, কম্পিউটারের সাহায্যে বড় বড় যন্ত্রপাতি নিয়ন্ত্রণে সব সময়ই A/D কনভার্টার ইনপুট ডিভাইস ব্যবহার করা হয়।

A/D কনভার্টারের আর একটি জনপ্রিয় প্রয়োগ স্থল হলো ভিডিও গেমস্ এর জয়স্টিক কন্ট্রোল (Joy Stick Control)। এই ব্যবস্থায় যিনি ভিডিও গেমস্ খেলছেন তিনি VDU স্ক্রিনের মধ্যে ক্যারেকটারের অবস্থানকে একটি গাড়ির গিয়ারের মতো দেখতে দণ্ডের সাহায্যে নিয়ন্ত্রন করতে পারেন।

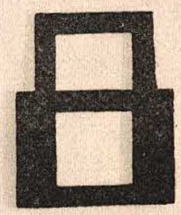
এছাড়া A/D কনভার্টারের আর একটি গুরুত্বপূর্ণ প্রয়োগ হলো মানুষের গলার আওয়াজ শুনে কম্পিউটারের প্রোগ্রাম নেওয়ার ব্যবস্থাতে। এই ব্যবস্থা যদিও প্রধানত এখনো গবেষনাগারের মধ্যেই আবদ্ধ হয়ে আছে, তবু আশা করা যায় আমরা পঞ্চম প্রজন্মের কম্পিউটারে এই ধরনের ইনপুট ডিভাইসের ব্যাপক প্রচলন দেখতে পাবো।

বার কোড

নিশ্চয়ই লক্ষ্য করেছেন বেশীর ভাগ বিদেশী বই, খাবারের ক্যান, ক্যাসেট, ফিল্মের বাক্স ইত্যাদির গায়ে সাদা আয়তক্ষেত্রাকার জায়গার ওপর বেশ অনেকগুলি সরু মোটা কালো ডোরাকাটা দাগ থাকে? এগুলিকে বলা হয় ইউ. পি. সি (U. P. C.)। U. P. C. কথাটি Universal Product Code কথাটির সংক্ষিপ্ত রূপ। অনেক সময় ইউ. পি. সিকে বার কোড (Bar Code) ও বলা হয়। এটি কম্পিউটারের একটি বিশেষ ইনপুট ব্যবস্থা।

বড় বড় বিভাগীয় বিপনিতে সমস্ত হিসেব নিকেশের কাজ স্বয়ংক্রিয় আর নির্ভুল ভাবে কম্পিউটারের মাধ্যমে করার জন্য এই বার কোড

ইনপুট ডিভাইস তৈরী করা হয়েছে। এই ব্যবস্থায় বিক্রয় সামগ্রির দাম, প্রস্তুত কর্তার নাম ইত্যাদি তথ্য, ওই বিক্রয় সামগ্রির প্যাকেটের গায়ে সরু মোটা কালো দাগের সংকেতের মাধ্যমে লেখা হয়। এই সাংকেতিক তথ্যকে পড়বার জন্য ব্যবহার করা হয় বার কোড রিডার যন্ত্র।



বার কোড

GB £ NET +002.25

ISBN 0-85934-079-1

£2.25

00225

9 780859 340793

চিত্র ২৬

ইংরিজি পেপারব্যাক বইয়ের গায়ে ছাপানো বার কোড।

IBM কম্পানির তৈরী সর্বাধুনিক বার কোড রিডার যন্ত্রের ওপর বার কোড যুক্ত কোন বিক্রয় সামগ্রি রাখলেই যন্ত্রের একটি ছোট জানলা থেকে বেরিয়ে আসে অল্প শক্তি সম্পন্ন লেসার (Laser) রশ্মি। যন্ত্রের অভ্যন্তরে, লেসার রশ্মির গতিপথে রাখা একটি ঘূর্ণায়মান চক্রে লাগানো একুশটি হলোগ্রাফিক লেন্স (Holographic Lens), ওই লেসার রশ্মিকে জানলার বাইরে পর্যায়ক্রমে একুশটি দূরত্বে এবং উচ্চতায় ফোকাস (Focus) এবং স্ক্যান (Scan) করে। ফলে বিক্রয় সামগ্রির প্যাকেটের আকৃতি যেমনই হোকনা কেন (সমতল কিম্বা বক্র), ওই একুশটি স্ক্যানের মধ্যে একটি স্ক্যান সব সময়ই বারকোডটিকে সঠিক ভাবে স্ক্যান করতে সক্ষম হয়। বারকোডটি লেসার রশ্মির দ্বারা সঠিক ভাবে স্ক্যান হওয়ার সময়, মোটা সরু কালো ডোরার জন্য প্রতিফলিত লেসার রশ্মিটির উজ্জ্বলতা (কোড অনুসারে) হ্রাস, বৃদ্ধি পায়। প্রতিফলিত রশ্মিটিকে হলোগ্রাফিক লেন্স এবং সাধারন লেন্সের মাধ্যমে ফোটো ডিটেকটরের ওপর ফেলা হয়।

ক্রমাগত পরিবর্তনশীল আলোর জন্য ফোটো ডিটেকটরে একটি বৈদ্যুতিক স্পন্দন সৃষ্টি হয়। এই বিদ্যুৎ স্পন্দন তারপর এনকোডার সার্কিটের মারফৎ ডিজিটাল সংকেতে রূপান্তরিত করে কম্পিউটারে পাঠানো হয়।

ডিজিটাল সংকেতটি পাবার পর কম্পিউটারের প্রধান গণনা কেন্দ্র



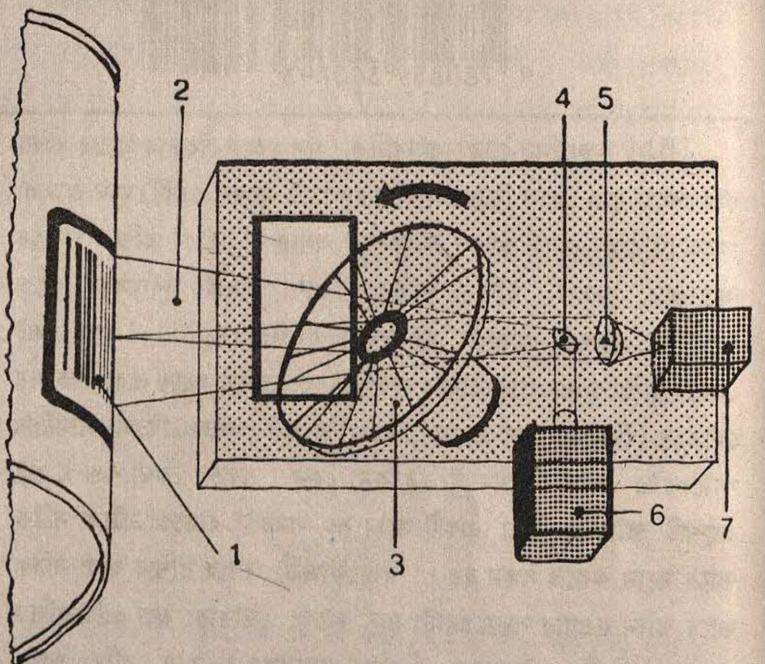
বার কোড

চিত্র ২৭

C. P. U. তার স্মৃতিভাণ্ডারে রক্ষিত সমস্ত বিক্রয় সামগ্রির বার কোড তথ্যের সঙ্গে নতুন পাওয়া তথ্যটি মিলিয়ে দেখে জেনে নেয় কোন জিনিসটি বিক্রি হতে চলেছে। তারপর ওই স্মৃতি ভাণ্ডারেই রক্ষিত বিক্রয় সামগ্রিটির নাম ও দাম জেনে নিয়ে কম্পিউটার বিল তৈরী করে দেয়, সেই সঙ্গে বিক্রয় সামগ্রিটির পরিবর্তিত নতুন স্টকটি কতো হলো সেটি স্মৃতি ভাণ্ডারে লিখে রাখে। অর্থাৎ একই সঙ্গে বিল তৈরী, স্টক আপটুডেট করা, হিসেব মেলানো, স্টক মেলানো ইত্যাদি ধরনের কাজগুলি সবই মুহূর্তে স্বয়ংক্রিয় এবং নির্ভুল ভাবে কম্পিউটার করে ফেলে।

আধুনিক বার কোড রিডার।

1—টিনের গায়ে ছাপানো বার কোড
2—লেসার রশ্মি
3—ঘূর্ণায়মান হলোগ্রাফিক লেন্স
4—অর্ধেক পারা লাগানো আয়না
5—সাধারন লেন্স
6—লেসার উৎস
7—ফোটো ডিটেকটর



পরিচ্ছেদ ৬

স্মৃতি সংরক্ষণ ব্যবস্থা

আভ্যন্তরীণ স্মৃতি ভাণ্ডার

কম্পিউটারে যা কিছু প্রোগ্রাম করা হয়, সেটি ইনপুট ডিভাইসের সাহায্যে বৈদ্যুতিক স্পন্দনে রূপান্তরীত হয়ে কম্পিউটারে প্রবেশ করে। এই বৈদ্যুতিক স্পন্দন সোজা চলে যায় কম্পিউটারের Internal Storage অর্থাৎ আভ্যন্তরীণ স্মৃতি ভাণ্ডারে। কম্পিউটার পরিভাষায় একে আবার অনেক সময় Primary Storage বা মূল স্মৃতি ভাণ্ডারও বলা হয়ে থাকে।

মূল স্মৃতি ভাণ্ডারের দুটি অংশ আছে। একটি অংশে ইনপুট ডিভাইস থেকে আগত সমস্ত প্রোগ্রাম জমা হয় এবং অপর অংশে আগে থেকে লিপিবদ্ধ থাকে কম্পিউটারের বিভিন্ন কার্যকলাপ নিয়ন্ত্রণের প্রোগ্রাম। দ্বিতীয় অংশের প্রোগ্রামটি কম্পিউটার নির্মাতারাই স্মৃতি ভাণ্ডারে ভরে দেন। এই প্রোগ্রামের সাহায্যে কম্পিউটার তার যাবতীয় কাজ কর্ম ঠিক ভাবে করার নির্দেশ পায়। যেমন ধরুন, আপনি কী বোর্ডের A বোতাম টিপলেন। অমনি কী বোর্ড থেকে A হরফের ডিজিটাল সংকেত স্মৃতি ভাণ্ডারের প্রথম অংশে জমা হলো। একই সঙ্গে দ্বিতীয় অংশে লিপিবদ্ধ প্রোগ্রাম কম্পিউটার কে নির্দেশ দিলো, 'VDU স্ক্রিণে A হরফ ফুটিয়ে তোল'। অথচ এই নির্দেশ আপনি কম্পিউটারে দেননি, নির্দেশটি স্মৃতি ভাণ্ডারের দ্বিতীয় অংশে লিপিবদ্ধ প্রোগ্রাম থেকে স্বয়ংক্রিয় ভাবেই চলে এলো। এই ভাবে আমাদের অজান্তে



আভ্যন্তরীণ স্মৃতি ভাণ্ডার

কম্পিউটারের সমস্ত খুঁটিনাটি কর্ম যাতে ঠিক ভাবে চলে তার যাবতীয় নির্দেশ স্মৃতি ভাণ্ডারের দ্বিতীয় অংশ থেকেই আসে।

স্মৃতি ভাণ্ডারের প্রথম অংশটিতে যেখানে আমাদের তৈরী প্রোগ্রাম জমা হয়, তাকে বলে RAM স্মৃতি ভাণ্ডার।

RAM কথাটি এসেছে Random Access Memory থেকে। RAM স্মৃতি ভাণ্ডারে আমরা যখন খুশি নতুন তথ্য সঞ্চয় করতে পারি, তথ্য পড়তে পারি আর ইচ্ছে করলে সব তথ্য মুছেও ফেলতে পারি। ইনপুট ডিভাইস থেকে আগত সমস্ত প্রোগ্রাম RAM স্মৃতি ভাণ্ডারে জমা হয়।

স্মৃতি ভাণ্ডারের দ্বিতীয় অংশ, যাতে কোম্পানী থেকে কম্পিউটার পরিচালনা এবং হাইলেভেল প্রোগ্রামিং ভাষা থেকে কম্পিউটার বোধ্য মেশিন ভাষায় অনুবাদ করার প্রোগ্রাম লিপিবদ্ধ থাকে, তাকে বলে ROM স্মৃতি ভাণ্ডার। Read Only Memory কথাটি থেকে ROM শব্দের উৎপত্তি। ROM স্মৃতি ভাণ্ডারে কোন ভাবেই নতুন প্রোগ্রাম লেখা যায় না। গানের রেকর্ডের মতো এই স্মৃতি থেকে কেবল লিপিবদ্ধ প্রোগ্রামই বার বার পড়া যেতে পারে। কম্পিউটারের পাওয়ার অফ করে দিলেও ROM এর তথ্য অবিকৃত থাকে। তাই ROM-কে Non Volatile স্মৃতি ব্যবস্থা বলা হয়।

RAM স্মৃতি কিন্তু ভোলাটাইল (Volatile), তাই কম্পিউটারের পাওয়ার অফ্ করলেই নিমেষের মধ্যে RAM-এ সঞ্চিত প্রোগ্রাম বা তথ্য মুছে যায় (যেমন ক্যালকুলেটারের মেমরি, এটিও একটি RAM স্মৃতি)।

মূল স্মৃতি ব্যবস্থায় ROM এবং RAM, উভয়েরই কিন্তু বিশেষ প্রয়োজন আছে। ROM স্মৃতিটি যদি RAM-এর মতো হতো, তাহলে কম্পিউটার অফ্ করলেই স্মৃতি মুছে যেতো। তারপর ফের কম্পিউটার চালু করলে ওই বিশেষ নির্দেশের অভাবে কম্পিউটার প্রোগ্রামের নির্দেশ বুঝতে না পেরে থেমে থাকতো। সেইজন্য ROM এর স্মৃতিকে চিরস্থায়ী করা হয়েছে এবং বাইরে থেকে ভরা প্রোগ্রামের তথ্যকে সঞ্চয় করার জন্য আলাদা RAM স্মৃতি ভাণ্ডারের ব্যবস্থা রাখা হয়েছে।

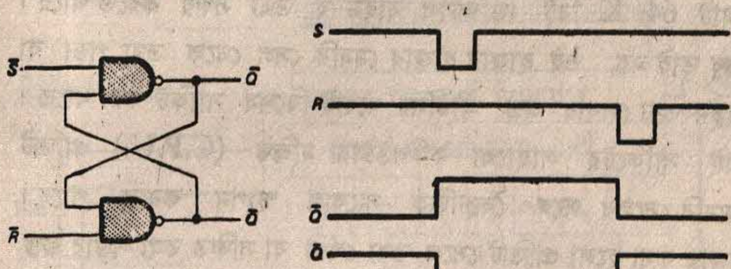
ফ্লিপ ফ্লপ



ফ্লিপ ফ্লপ

চিত্র ২৮

কম্পিউটারের আভ্যন্তরীণ স্মৃতি ভাণ্ডারে ডিজিটাল বিদ্যুৎ স্পন্দন দিয়ে তৈরী বাইনারি তথ্য ইলেক্ট্রনিক সার্কিটের সাহায্যে সঞ্চয় করে রাখা হয়। যে ইলেক্ট্রনিক সার্কিট (সর্বপ্রথম এবং এখনও) এই কাজে ব্যবহার করা হয়,তার নাম ফ্লিপ ফ্লপ (Flip Flop)। ফ্লিপ ফ্লপ আসলে একটি ইংরিজি ধ্বন্যাত্মক শব্দ, যেমন বাংলায় খুট খাট, টুপ টাপ ইত্যাদি। শব্দটি শুনলেই কেমন যেন মনে হয় কোন ইলেকট্রিক সুইচের অন্ অফ্ করার শব্দ তাই না? ফ্লিপ ফ্লপ সার্কিটটিও কিন্তু প্রায় সেই রকমেরই একটি সার্কিট। এই সার্কিটে দুটি ইনপুট আর দুটি আউটপুটের তার থাকে। ইনপুট তার দুটির একটিতে বাইনারি 1 এবং অন্যটিতে বাইনারি 0 লেভেলের ভোল্টেজ প্রয়োগ করলে আউটপুট দুটিতেও বাইনারি 1 এবং বাইনারি 0 লেভেলে ডিজিটাল আউটপুট পাওয়া যায়। এইবার যদি ইনপুটের দুটি তারে ডিজিটাল ভোল্টেজ উল্টে দেওয়া হয়, অর্থাৎ আগে যেখানে 0 ছিল সেখানে 1 এবং যেখানে 1 ছিলো সেখানে 0 সিগন্যাল দেওয়া হয়, তবে আউটপুটের দুটি তারেরও ডিজিটাল আউটপুট উল্টে যায়। সুতরাং ইনপুট তারের সাহায্যে ফ্লিপ ফ্লপের আউটপুটকে বাইনারি 1 বা 0 স্টেটে ইচ্ছেমত সেট্ করা যায়। ফ্লিপ ফ্লপের বৈশিষ্ট্য হলো, আউটপুটকে নির্দিষ্ট স্টেটে নিয়ে আসার পর যদি ইনপুট তার দুটি থেকে বিদ্যুৎ সংযোগ



(ক) (খ)

(ক) দুটি NAND গেট সার্কিট দিয়ে তৈরী ফ্লিপ ফ্লপ সার্কিট

(খ) R, S ইনপুটের ভোল্টেজ অনুসারে আউটপুটের ভোল্টেজ পরিবর্তন



আভ্যন্তরীণ স্মৃতি STATIC RAM

বিচ্ছিন্ন করা হয়, আউটপুটের ডিজিটাল সিগন্যাল কিন্তু তার আগের স্টেটেই অপরিবর্তিত থেকে যায়। যতক্ষণ না আবার ইনপুট তারে উল্টো সিগন্যাল পাঠিয়ে আউটপুট সিগন্যাল দুটিকে ওল্টানো হচ্ছে, ততক্ষণ তারা ওই একই অবস্থায় বা স্টেটে (State) থাকে। ফ্লিপ ফ্লপ সার্কিটের এই বিশেষ বৈশিষ্ট্যকেই ডিজিটাল তথ্য সংরক্ষণের কাজে ব্যবহার করা হয়।

ফ্লিপ ফ্লপের একটি ইনপুটকে বলা হয় Set বা সংক্ষেপে S ইনপুট এবং অপরটির নাম Reset বা সংক্ষেপে R ইনপুট। আউটপুট দুটিকে যথাক্রমে X এবং X′ (X এর কমপ্লিমেণ্ট) বলা হয়। আমরা সব সময় X আউটপুটকেই ফ্লিপ ফ্লপ আউটপুট বলে গণ্য করি। এই ধরনের ফ্লিপ ফ্লপকে R S ফ্লিপ ফ্লপও বলা হয়।

আধুনিক ইলেকট্রনিক TTL মেমরি IC-র মধ্যে এই ধরনের অতিক্ষুদ্র ফ্লিপ ফ্লপ প্রচুর পরিমানে তৈরী করে ডিজিটাল স্মৃতি সরক্ষণের কাজে ব্যবহার করা হয়। প্রতিটি ফ্লিপ ফ্লপ কেবল মাত্র একটি ডিজিটাল 1 কিম্বা ডিজিটাল 0 সংকেত লিপিবদ্ধ করতে পারে। কম্পিউটার পরিভাষায় এই সর্বনিম্ন তথ্যকে এক বিট্ (Bit) তথ্য আর প্রতিটি ফ্লিপ ফ্লপকে একক স্মৃতি কোষ (Memory Cell) বলা হয়।

কম্পিউটারের স্মৃতি ধারণ ক্ষমতা প্রকাশ করা হয় **Kilo Byte** (কিলো বাইট বা সংক্ষেপে শুধু **K** এককে। যেমন 16**K** অথবা **48 K** ইত্যাদি ; **1 Byte** বলতে বোঝায় **4 Bit** তথ্য এবং **1 K Byte** বলতে বোঝায় 2^{10} **Bit** অর্থাৎ **1024 Bit** তথ্য সুতরাং সেই হিসেবে **48 K** স্মৃতি ধারণ ক্ষমতা সম্পন্ন একটি কম্পিউটারের আভ্যন্তরীণ স্মৃতি ভাণ্ডার **49152 Bit** ডিজিটাল তথ্য সংরক্ষণ করতে পারে।

আধুনিক Static RAM স্মৃতির গঠন

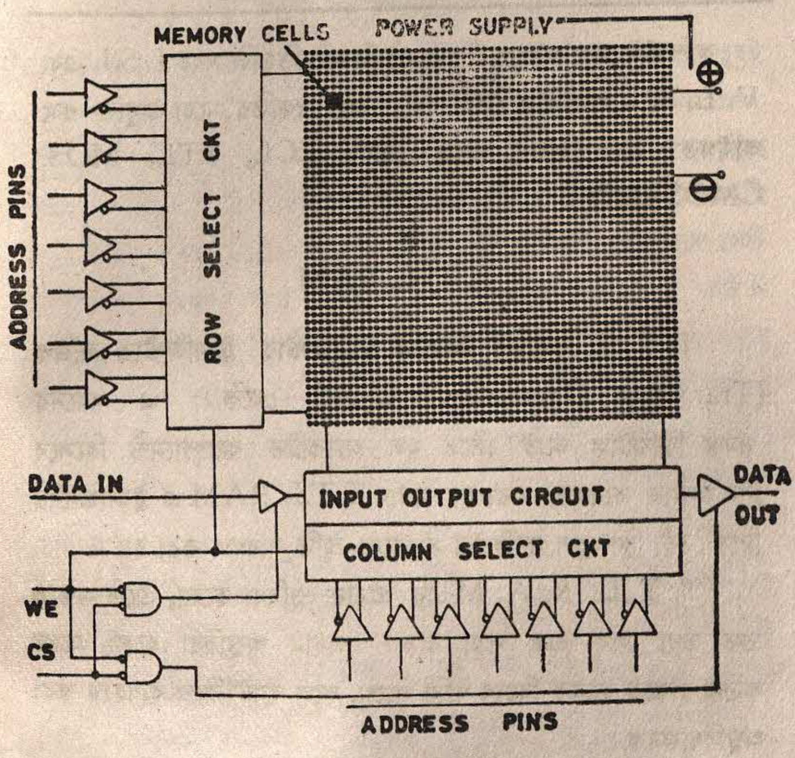
বর্তমানে L.S.I. (Large Scale Integration) এবং V L.S.I. টেকনোলজির দৌলতে একটি ছোট আইসির মধ্যেই 65536টি ফ্লিপ ফ্লপ সার্কিট ঢুকিয়ে দেওয়া সম্ভব হয়েছে! এই 65536টি ফ্লিপ ফ্লপ মোট 64 K বিট্ ডিজিটাল বাইনারি তথ্য সঞ্চয় করতে পারে। শুধু তাই নয়, ওই হাজার হাজার মেমরি সেল থেকে তথ্য পড়া বা নতুন তথ্য লেখার জন্য আইসির মধ্যেই বিশেষ সার্কিট করা থাকে। এই সার্কিটের সাহায্যে কম্পিউটার মস্তিষ্ক (C.P.U.) প্রতিটি মেমরি সেলের সঙ্গে বৈদ্যুতিক সংযোগ স্থাপন করতে পারে। মজার কথা হলো প্রতিটি সেলে তথ্য লেখা বা সঞ্চিত তথ্য পড়ার জন্য কিন্তু হাজার হাজার আলাদা তারের প্রয়োজন হয় না। এই কাজের সুবিধার জন্য আই-সির অভ্যন্তরে মেমরি সেল গুলিকে ম্যাট্রিক্স

(Matrix) নিয়মে সাজানো হয়। ম্যাট্রিক্স বলতেবোঝায় গ্রাফ পেপারের চৌকো খোপের মতো আলাদা আলাদা সারি (Row) এবং স্তম্ভে (Column) সাজানোর ব্যবস্থা। X এবং Y কোঅর্ডিনেট জানা থাকলে যেমন গ্রাফ পেপারে যে কোন বিন্দু বসানো (Plot) যায়, ঠিক তেমনি মেমরি সেলের স্তম্ভ এবং সারিতে বিদ্যুৎ সংযোগ করে তাদের সংযোগ বিন্দুতে অবস্থিত সেলের সঙ্গে বাইরের সার্কিটের বৈদ্যুতিক স্থাপন করা হয়। এই কাজের জন্য সমস্ত মেমরি আই সি-তেই X এবং Y ডিকোডার ইলেকট্রনিক ব্যবস্থা থাকে। আই সির অ্যাড্রেস পিনের সঙ্গে এই ডিকোডার সার্কিটের সংযোগ থাকে। তাছাড়া থাকে রিড্/রাইট্ এনেবল্ (Read-Write enable) এবং ডেটা (Data) পিন। যখন কোনো RAM স্মৃতি থেকে তথ্য পড়তে হয় তখন রিড্-রাইট্ এনেবল্ পিনে বিশেষ ভোল্টেজ পাঠিয়ে মেমরি সেলগুলি থেকে তথ্য পড়বার প্রস্তুতি নেওয়া হয়। তারপর অ্যাড্রেশ(Address)



আভ্যন্তরীণ স্মৃতি STATIC RAM

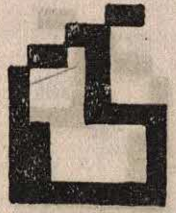
চিত্র ২৯



আধুনিক Static RAM এর আভ্যন্তরীণ সার্কিট বিন্যাস

Row Select Column Sebct এর সঙ্গে সংযুক্ত **Address** পিন গুলিতে ডিজিটাল সংকেত পাঠিয়ে **Matrix** হিসেবে সাজানো **Memory Cell** গুলি থেকে যে কোনো একটি বা একাধিক **Memory Cell** (বা **Flip Flop**) এর সঙ্গে **Data In** কিম্বা **Data Out** এর বৈদ্যুতিক সংযোগ ঘটানো যায়।

WE (Write Enable) **Pin** এর সাহায্যে ফ্লিপ ফ্লপ থেকে ইচ্ছে মতো তথ্য লেখা **(Writ)** কিম্বা পড়া **(Read)** যায়।



আভ্যন্তরীণ স্মৃতি

পিনের সাহায্যে XY ডিকোডার সার্কিটে সংকেত পাঠিয়ে কোন বিশেষ সারি এবং স্তম্ভের সংযোগ বিন্দুতে স্থাপিত একটি বা একাধিক মেমরি সেলকে অ্যাড্রেশ করার (বেছে নেওয়ার) তথ্য পাঠানো হয়। সঙ্গে সঙ্গে ডেটা পিনে ওই মেমরি সেলে সঞ্চিত বাইনারি তথ্যের ডিজিটাল ভোল্টেজ পাওয়া যায়।

RAM স্মৃতিতে কোনো নতুন তথ্য লিখতে হলে প্রথমে রিড-রাইট্ এনেবল্ পিনে পূর্বের ঠিক উল্টো ভোল্টেজ পাঠিয়ে মেমরি সেলে তথ্য লেখার প্রস্তুতি নেওয়া হয়। তারপর আগের মতো অ্যাড্রেশ পিনের মাধ্যমে আমাদের পছন্দসই মেমরি সেল (যাকে কম্পিউটার পরিভাষায় বলে Memory Location) অ্যাড্রেশ করে যে তথ্য লিপিবদ্ধ করতে হবে, তাকে ডাটা পিনে প্রয়োগ করলেই ওই তথ্য বিশেষ মেমরি সেলে লেখা হয়ে যায়। সুতরাং ডেটা পিনের মাধ্যমে আমরা যেমন সঞ্চিত তথ্য পড়তে পারি, তেমনি ওই একই পিন দিয়ে যে তথ্য সঞ্চয় করতে হবে, তাকে আই সিতে পাঠানো হয়।

কম্পিউটারে যে পরিবাহী তারের মাধ্যমে তথ্য যাতায়াত করে, তাকে বলা হয় ডেটা বাস (Data Bus)। ডেটা বাস এ দ্বিমুখী প্রবাহ চলে বলে একে বাই ডাইকেরশনাল বাস (Bi Directional Bus) ও বলা হয়।

মেমরি সেল অ্যাড্রেশ করার সংকেত বহনকারী তার গুচ্ছর নাম হয় অ্যাড্রেশ বাস এবং কম্পিউটারের বিভিন্ন অংশকে নিয়ন্ত্রণ করার নির্দেশ যে তার গুচ্ছের মাধ্যমে প্রবাহিত হয় তাকে বলে কন্ট্রোল বাস Control Bus।

বিভিন্ন ধরনের RAM স্মৃতি

কম্পিউটারের আভ্যন্তরীণ ইলেক্ট্রনিক স্মৃতিগুলি সবই L.S.I. এবং V. L. S. I. পদ্ধতিতে তৈরী হয়। এই পদ্ধতির মধ্যে প্রযুক্তি এবং কার্যগত নানা বিভাগ আছে যেমন ECL, TTL, MOS, CMOS প্রভৃতি।

T. T. L. RAM

T. T. L কথাটির উৎপত্তি ট্রানজিস্টার ট্রানজিস্টার লজিক (Transistor Transistor Logic) থেকে। এ ধরনের সমস্ত ডিজিটাল আই সিতে মূল অ্যাকটিভ কম্পোনেন্ট হিসেবে ট্রানজিস্টার ব্যবহার করা হয় অর্থাৎ T.T.L RAM-এ ট্রানজিস্টার দিয়ে তৈরী ফ্লিপ ফ্লপ সার্কিটের সাহায্যে স্মৃতি সংরক্ষণ করা হয়।

T. T .L. R. A. M-এর প্রধান সুবিধা হলো, এতে অত্যন্ত দ্রুত তথ্য লেখা এবং পড়া যায়। প্রধান অসুবিধা হলো, এদের সক্রিয় রাখতে অনেক বিদ্যুৎ শক্তি লাগে, ফলে ব্যাটারিতে ব্যবহার করা অসুবিধাজনক।

CMOS RAM

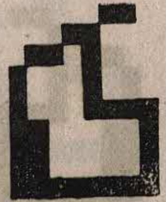
CMOS অর্থাৎ কমপ্লিমেনটারি মেটাল অক্সাইড সেমিকণ্ডাকটর (Complementory Metal Oxide Semiconductor) RAM-ও খুব জনপ্রিয়। CMOS RAM, TTL RAM-এর থেকে অপেক্ষাকৃত ধীর গতিতে কাজ করলেও এদের বিদ্যুৎ চাহিদা অত্যন্ত কম। তাই ব্যাটারি লাগানো কম্পিউটারের মূল পাওয়ার সাপ্লাই অফ্ করে দিলেও কম্পিউটারের অভ্যন্তরে লাগানো নিকেল ক্যাডমিয়াম রিচার্জেবল বাটন সেল এদের বহু দিন সক্রিয় রাখতে পারে। ফলে কম্পিউটার অফ্ করে দিলেও প্রোগ্রাম করা তথ্য কম্পিউটারের RAM স্মৃতিতে লিপিবদ্ধ থেকে যায়। পরে আবার যখন কম্পিউটার চালানো হয় তখন অনায়াসে দেখে নেওয়া যায় শেষ প্রোগ্রাম কি ছিলো। ইদানিং সমস্ত পকেট পারসোনাল কম্পিউটারেই CMOS RAM ব্যবহার করা হচ্ছে।



আভ্যন্তরীণ স্মৃতি DRAM

DRAM

র‍্যামের একটি নতুন প্রজন্ম ক্রমশ কম্পিউটার জগতে খুবই জনপ্রিয় হয়ে উঠছে। এই প্রজন্মের নাম DRAM। Dynamic RAM-কে সংক্ষিপ্ত করে ড্র্যাম কথাটির উৎপত্তি। এ ধরনের র‍্যামে ফ্লিপ ফ্লপের বদলে MOS ট্রানজিস্টর আর ক্যাপাসিটর দিয়ে তৈরী মেমরি সেল ব্যবহার করা হয়। মেমরি সেলের ক্যাপাসিটর চার্জ হলে সেটি বাইনারি 1 বোঝায় আর ক্যাপসিটরে কোনো চার্জ না থাকলে সেটি 0 বোঝায়। সাধারণ ফ্লিপ ফ্লপের থেকে এ ধরনের মেমরি সেল আকারে অনেক ছোটো আর দামে অনেক সস্তা। তাই একটি সাধারণ র‍্যামের মাপের তৈরী আই সি প্যাকে অনেক বেশী DRAM মেমরি সেল পুরে দেওয়া যায়। তবে DRAM-এর স্মৃতি ব্যবস্থা TTL RAM-এর মতো স্ট্যাটিক (Static) নয়, এটি ডায়নামিক (Dynamic)। ফ্লিপ ফ্লপ যুক্ত স্ট্যাটিক RAM এ কোনো তথ্য লেখার পর সেই তথ্য যখন খুশি মেমরি অ্যাড্রেশ করে পড়া যায়। এতে লিখিত তথ্য মুছে যায় না। কিন্তু DRAM-এ লিখিত তথ্য পড়বার সময় মেমরি সেল ক্যাপাসিটরকে ডিসচার্জ করতে (discharge) হয়। ফলে একবার তথ্য পড়লেই মেমরি সেলের ক্যাপাসিটর চার্জহীন



আভ্যন্তরীণ স্মৃতি ROM

হয়ে সমস্ত তথ্য মুছে যায়, আর সব সেলে বাইনারি 0 তথ্য লিখিত হয়ে যায়! তাছাড়া DRAM মেমরি সেলের আনুবিক্ষণীক ক্যাপাসিটর গুলি এতোই ছোট, যে তাদের চার্জ করার কয়েক মিলি সেকেণ্ডের মধ্যে তারা চার্জ হারিয়ে ফেলে। এই দুটি অসুবিধার হাত থেকে রেহাই পেতে DRAM Controller সার্কিটের সাহায্যে কয়েক মিলি সেকেণ্ড অন্তর DRAM স্মৃতির সব কটি মেমরি লোকেশন করে একের পর এক অ্যাড্রেশ করে তাতে লিখিত তথ্যকে পুনরায় লেখা হয়। একে কম্প্যিউটার পরিভাষায়. বলা হয় মেমরি রিফ্রেস (Memory Refresh) করা। এই ব্যাপারটি DRAM স্মৃতির এক মাত্র অসুবিধা। তবে বেশীর ভাগ কম্পিউটারে এটি খুব বড় অসুবিধার কারণ হয়ে দাঁড়ায় না, কারণ কেবল মাত্র একটি মেমরি কনট্রোলার আই সি-র সাহায্যে এক সঙ্গে অনেক গুলি DRAM রিফ্রেস যায়।

কম দাম এবং ছোটো জায়গায় অনেক বেশী স্মৃতি ধারণ ক্ষমতা DRAM-কে আজকাল খুবই জনপ্রিয় করে তুলছে। ইদানিং বহু মাইক্রো কমপিউটারে সাধারণ র‍্যামের বদলে ড্র্যাম ব্যবহার করা হচ্ছে।

রিড ওনলি মেমরি (Read Only Memory)

রিড ওনলি মেমরি (Read Only Memory) কথাটি সংক্ষিপ্ত করে ROM শব্দটি এসেছে। ROM ইলেকট্রনিক স্মৃতির বৈশিষ্ট্য হলো, এতে কেবলমাত্র একবারের জন্য কোন একটি তথ্য লেখা যায় আর একবার লেখা হয়ে গেলে স্মৃতির ওই তথ্য চিরদিনের জন্য লিপিবদ্ধ হয়ে যায়। ইলেকট্রনিক স্মৃতির পাওয়ার সাপ্লাইয়ে ছেদ ঘটলে কিম্বা দীর্ঘ দিন তাকে এমনি ফেলে রাখলেও ROM-এ লিপিবদ্ধ স্মৃতির কোনো পরিবর্তন ঘটে না। সুতরাং ROM এর সঙ্গে গানের রেকর্ডের তুলনা করা যেতে পারে।

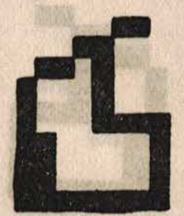
ROM গুলি সবই ছোটো ছোটো আই সি প্যাকে পাওয়া যায়। এতে বেশ কতগুলি ইনপুট আর আউটপুট লাইন থাকে। ইনপুট লাইনের সাহায্যে ROM-এর বিভিন্ন মেমরি লোকেশন অ্যাড্রেশ করা হয় আর আউটপুট লাইনে সেই আড্রেশ লোকেশনে সংরক্ষিত বাইনারি তথ্য পাওয়া যায়।

শুধু কম্পিউটারেই নয়, আমাদের দৈনন্দিন জীবনেও ROM-

এর প্রয়োগ আমরা হামেশাই দেখতে পাই। মিউজিক্যাল ইলেক্‌ট্রনিক কলিং বেল, এবং ভি.ডি.ও গেম, এ দুটো জিনিসেরই মূল ইলেকট্রনিক অংশটি হলো ROM। ইলেকট্রনিক আই সি নির্মাতারাই ROM তৈরী করার সময়ে তার বিভিন্ন মেমরি সেল-এ প্রয়োজনীয় বাইনারি তথ্য ঢুকিয়ে দেন। ROM-এ তথ্য পরিবর্তনের দরকার নেই বলে ROM মেমরি সেল গুলি তৈরী করা আরো সহজ কেবলমাত্র একটি ফিউজ কিম্বা ডায়োড দিয়ে এই সেল তৈরী করা হয়। যে সেলে বাইনারি 1 তথ্য রাখতে হবে, সেই সেলে ফিউজ বা ডায়োডকে পরিবাহী হিসেবে রাখা হয়। ফলে ওই সেলকে অ্যাড্রেশ করা হলে পরিবাহী ফিউজ বা ডায়োডের মাধ্যমে আউটপুটে বাইনারি 1 তথ্য পাওয়া যায়। আর যে সব সেলে বাইনারি 0 রাখতে হবে, বিশেষ যন্ত্রের সাহায্যে বাইরে থেকে সেইসব সেলের ফিউজকে পুড়িয়ে দেওয়া হয়, অথবা ROM তৈরী করার সময় সেসব সেলে ডায়োড লাগানো হয় না। সুতরাং ওই সেলগুলি অ্যাড্রেশ করলে, কোনো ফিউজ বা ডায়োড না থাকার দরুণ আউটপুটে 0 তথ্য পাওয়া যায়। কম্পিউটারে বিভিন্ন কাজের জন্য বিভিন্ন কোম্পানি তাদের নিজস্ব নানা রকম ROM বাজারে বিক্রি করেন।

PROM

সাধারণ ROM এর প্রধান অসুবিধা হলো, ব্যবহারকারী এতে নিজের ইচ্ছে মতো প্রোগ্রাম লিপিবদ্ধ করতে পারেন না। অথচ বাজারে যে সমস্ত প্রোগ্রাম করা ROM পাওয়া যায়, তাতে সমস্ত কাজ হয়না। তাই এসব ক্ষেত্রে ব্যবহার করা হয় PROM বা প্রোগ্রামেবল্ রিড অনলি মেমরি। নতুন অবস্থায় এই ইলেকট্রনিক স্মৃতির সমস্ত মেমরি সেলে একটি করে ফিউজ লাগানো থাকে। ফলে সদ্য কেনা একটি প্রম-এর সমস্ত মেমরি লোকেশনেই 1 তথ্য পাওয়া যায়। বিশেষ প্রম প্রোগ্রামার যন্ত্রের সাহায্যে ব্যবহারকারী প্রমের বিশেষ বিশেষ মেমরি সেলের ফিউজ বিদ্যুৎ প্রবাহ চালিয়ে পুড়িয়ে দিয়ে সেটিতে 1 এবং 0 ইচ্ছে মতো প্রোগ্রাম করে নিতে পারেন। পুরো প্রোগ্রামিং শেষ হলে জিনিসটি একটা ROM এ পরিণত হয়।



আভ্যন্তরীণ স্মৃতি PROM

কোম্পানী থেকে প্রোগ্রাম করে দেওয়া ROMকে বলা হয় Preprogrammed ROM (যেমন গান টেপ করা ক্যাসেটকে বলা হয় Prerecorded Cassette)। বিভিন্ন কাজের জন্য এবং মুখ্যত নানা ধরনের ভি. ডি. ও গেম কম্পিউটারের মাধ্যমে খেলার জন্য এই ROM বিক্রি করা হয়। বর্তমানে ভারতের তিনটি প্রধান শহরে (বোম্বে, মাদ্রাজ এবং দিল্লি) কম্পিউটার উৎসাহিদের জন্য ROM লাইব্রেরী খোলা হয়েছে। এই লাইব্রেরীর সদস্যরা মাসিক চাঁদার বিনিময়ে বিভিন্ন প্রোগ্রাম যুক্ত ROM, ফ্লপি ডিস্ক এবং ক্যাসেট কিছুদিনের জন্য বাড়িতে ব্যবহার করতে আনতে পারেন।



আভ্যন্তরীন স্মৃতি EPROM

অধিক সংখ্যক PROM একই প্রোগ্রামে প্রোগ্রামিং করার জন্য বিশেষ প্রম প্রোগ্রামার যন্ত্র পাওয়া যায়। এই প্রোগ্রামার যন্ত্রটি পাঞ্চ কার্ড, পাঞ্চ টেপ বা ম্যাগনেটিক টেপে সংরক্ষিত প্রোগ্রাম অনুসারে স্বয়ংক্রিয়ভাবে PROM প্রোগ্রাম করে দেয়।



(ক)



(খ)

চিত্র ৩০

(ক) ইলেকট্রনিক PROM প্রোগ্রামার যন্ত্র

(খ) EPROM আই. সি. কে প্রোগ্রাম করার জন্য PROM প্রোগ্রামের যন্ত্রের ZIF সকেটে বসানো হচ্ছে

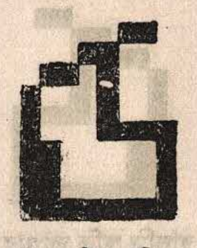
EPROM

PROM সাধারণত ছোটো আর মাঝারি ইলেকট্রনিক যন্ত্র নির্মাতারাই ব্যবহার করেন, কারণ PROM একবার প্রোগ্রাম করা হয়ে গেলে তার তথ্য পরিবর্তন করা যায় না। যদি একটি মেমরি লোকেশনও অসাবধানতা বশত ভুল প্রোগ্রাম হয়ে যায়, তবে পুরো PROMটিই বরবাদ। তাই সাধারণ কম্পিউটার ব্যবহারকারী এবং কম্পিউটার নির্মাতাদের জন্য একটি বিশেষ ধরণের ROM তৈরী হয়, এটির নাম এপরম EPROM বা Erasble Programable ROM (ইরেজেবল প্রোগ্রামেবল রিড ওনলি মেমরি)। কম্পিউটার জগতে সব থেকে জনপ্রিয় ROM হলো এই EPROM।

EPROM-এর সঙ্গে সাধারণ ক্যাসেটের কিছুটা তুলনা করা চলে। একটি ফাঁকা ক্যাসেটে যেমন গান রেকর্ড করা যায়, তেমনি একটি ফাঁকা EPROM এ তথ্য লিপিবদ্ধ করা যায়। রেকর্ডিং-এর পর ক্যাসেটকে যেমন যতবার খুশি বাজিয়ে গান শোনা যায়, EPROM থেকেও তেমনি যতবার খুশি লিপিবদ্ধ তথ্য পড়া যায়। আবার ক্যাসেটের গান

যেমন ইচ্ছে করলেই মুছে ফেলা চলে, এপরমকেও তীব্র আলট্রা ভায়োলেট আলোর তলায় পনেরো থেকে তিরিশ মিনিট রেখে দিলে তাতে রেকর্ডিং করা পুরানো তথ্য একদম মুছে যায়। এরপর তাতে আবার নোতুন তথ্য লিপিবদ্ধ করা চলে।

ROM-এর লিপিবদ্ধ তথ্য পুনরায় মুছে ফেলার সুবিধাই EPROM-জনপ্রিয়তার প্রধান কারণ। একটি EPROM-এর আকৃতি হয় মাঝারি সাইজের আই সির মতো। তবে সাধারণ আই সির সঙ্গে এপরমের প্রধান তফাৎ হলো, এতে একটি ছোটো কোয়ার্জের স্বচ্ছ জানলা থাকে। ওই জানলা দিয়ে আই সির সিলিকন ওয়েফার দেখা যায়। তথ্য মুছে ফেলার সময়ে একটি আলট্রা ভায়োলেট ল্যাম্পের তলায় EPROMকে রাখলে, ল্যাম্পের আলট্রা ভায়োলেট রশ্মি ওই জানলা দিয়ে সিলিকন ওয়েকারের ওপর পড়ে, আর কিছুক্ষণের মধ্যেই EPROM-এর তথ্য মুছে যায়।



আভ্যন্তরীণ স্মৃতি
EEPROM, EAPROM

চিত্র ৩১



আধুনিক EEPROM আই. সি

EEPROM এবং EAPROM

EPROM এতো জনপ্রিয় হওয়া সত্ত্বেও তার প্রধান অসুবিধা হলো স্মৃতির তথ্য মুছতে গেলে তাকে আলট্রা ভায়োলেট আলোতে আধ ঘণ্টা মতো রাখতে হয়। এই কাজটি কখনোই EPROM-কে কম্পিউটারে বসানো অবস্থায় করা যায় না। পুরোনো তথ্য মুছে নোতুন তথ্য লিখতে হলে সর্বদাই সার্কিট থেকে EPROMকে খুলে নিতে হয়। এটি সাধারণ কম্পিউটার ব্যবহারকারীর কাছে খুব একটি সুবিধাজনক ব্যবস্থা নয়। তাই সহজে পুরোনো তথ্য মুছে ফেলে নোতুন তথ্য প্রোগ্রাম করার ব্যবস্থা করা হয়েছে EEPROM আর EAPROM ধরণের স্মৃতিতে। EAPROM শব্দটি এসেছে Electrically Alterable PROM আর EEPROM এসেছে Electrically Erasable PROM কথা থেকে। EAPROM স্মৃতির তথ্য বিদ্যুৎ প্রবাহ পাঠিয়ে মুছে ফেলা যায়, আর EEPROMকে কম্পিউটারে বসানো অবস্থাতেই মুছে ফেলে নোতুন প্রোগ্রামিং করা যায়।



বাড়তি স্মৃতি ভাণ্ডার

বাড়তি স্মৃতি ভাণ্ডার

আমরা যেমন অনেক কথা বা তথ্য, মনে না রেখে, ভবিষ্যতে কাজে লাগবে বলে খাতায় নোট করে নিই ; ঠিক তেমনি কম্পিউটারের তথ্য জমিয়ে রাখার জন্য বাড়তি স্মৃতি ভাণ্ডারের (External Memory) দরকার হয়। এই বাড়তি স্মৃতি ভাণ্ডার, কম্পিউটারের সঙ্গে বাইরে থেকে তার দিয়ে সংযুক্ত করা যায়। উপযুক্ত নির্দেশ পেলেই কম্পিউটার আভ্যন্তরীণ স্মৃতি ভাণ্ডারে সংরক্ষিত তথ্যকে ওই বাড়তি স্মৃতি ভাণ্ডারে লিখে রাখে। এতে সুবিধা হলো, কম্পিউটারকে অফ্ করে রাখার সময়ে তার আভ্যন্তরীণ RAM স্মৃতিতে রাখা প্রোগ্রাম মুছে গেলেও সেই তথ্য তখন বাড়তি স্মৃতি ভাণ্ডারে যথাযথ ভাবে সংরক্ষিত থাকে। পরে যখনই আবার কম্পিউটারকে ওই প্রোগ্রাম করার দরকার পড়ে তখন স্রেফ কম্পিউটারকে নির্দেশ দিতে হয় 'বাড়তি স্মৃতি থেকে অমুক তথ্য পড়ো'। অমনি কম্পিউটার মুহূর্তের মধ্যে বাড়তি স্মৃতি ভাণ্ডার থেকে সঠিক তথ্য খুঁজে নিয়ে সেই প্রোগ্রাম তার আভ্যন্তরীণ RAM-এ ভরে নেয়, আর সেই প্রোগ্রাম অনুসারে কাজ করার জন্য তৈরী হয়ে যায়।

তবে একটা কথা সব সময়ই মনে রাখা উচিত, বাড়তি স্মৃতি ভাণ্ডার কিন্তু কম্পিউটারের আভ্যন্তরীণ স্মৃতি ভাণ্ডারের পরিপূরক নয়। বাড়তি স্মৃতি ভাণ্ডার থেকে কখনোই RAM স্মৃতির মতো দ্রুত তথ্য লেখা, বা লিখিত তথ্য পড়া যায় না। সুতরাং যে সব কম্পিউটারের আভ্যন্তরীণ স্মৃতি ভাণ্ডার সীমিত, সেই কম্পিউটারে যত বড়ই বাড়তি স্মৃতি ভাণ্ডার লাগানো হোক না কেন, তাতে কখনোই বড় প্রোগ্রাম চালানো যায় না। বাড়তি স্মৃতি ভাণ্ডার হলো আমাদের নোট বইয়ের মতো। কারুর স্মৃতিশক্তি যদি দুর্বল হয় তবে সেকি একটা বড় বই পড়ে, মনে রাখতে পারে? সীমিত আভ্যন্তরীণ স্মৃতিযুক্ত কম্পিউটারেও এই ব্যাপারটি ঘটে।

বর্তমানে বহু ধরনের বাড়তি স্মৃতি ভাণ্ডার কিনতে পাওয়া যাচ্ছে। এদের মধ্যে সিংহ ভাগেই, চৌম্বক পদার্থের প্রলেপ লাগানো ফিতে কিম্বা চাকতি, ডিজিটাল তথ্য লিপিবদ্ধ করার কাজে ব্যবহার করা হয়। এছাড়া বেশ কিছু সলিডস্টেট স্মৃতি ভাণ্ডারও পাওয়া যায়। এ সব সলিডস্টেট স্মৃতিতে প্রধানত EPROM, Bubble Memory,

C.C.D.Memory ইত্যাদি ধরনের আই সি ব্যবহার করা হয়ে থাকে। তবে চৌম্বক পদ্ধতিতে স্মৃতি সংরক্ষণ ব্যবস্থাই কম্পিউটার ব্যবহারকারী দের মধ্যে সব থেকে জনপ্রিয়। এই ধরনের স্মৃতি ব্যবস্থায়, স্মৃতি সংরক্ষণ বাবদ খরচা অন্যান্য ব্যবস্থার থেকে কম পড়ে।



চৌম্বক পদার্থে তথ্য সংরক্ষণ

চৌম্বক পদার্থে তথ্য সংরক্ষণ পদ্ধতি

চৌম্বক পদার্থে ডিজিটাল তথ্য সংরক্ষণের মূল পদ্ধতির সঙ্গে সাধারণ টেপ রেকর্ডারের কার্যপ্রণালীর খুব মিল আছে। টেপ রেকর্ডারে যেভাবে টেপ হেড কয়েলের মধ্যে বিদ্যুৎ স্পন্দন পাঠিয়ে, হেডের সামনে দিয়ে চালিত চৌম্বক ফিতেতে বিভিন্ন রকম চৌম্বক ক্ষেত্র সৃষ্টি করে গান রেকর্ড করা হয়, কম্পিউটারের চৌম্বক স্মৃতি ভাণ্ডারের কাজও ওই একই রকম। তবে এক্ষেত্রে তথ্য রেকর্ড করতে বিশেষ ধরনের বিদ্যুৎ স্পন্দন ব্যবহার করা হয়।

আধুনিক টেপ রেকর্ডারে যে শব্দ রেকর্ডিং করতে হয়, সেই শব্দ তরঙ্গের হুবহু সম আকৃতির বিদ্যুৎ তরঙ্গ একটি উচ্চকম্পাঙ্কের সাইন ওয়েভ (Bias Frequency) তরঙ্গের সঙ্গে মিশিয়ে টেপ হেডে পাঠানো হয়। এই মিশ্রিত তরঙ্গের প্রভাবে টেপ হেড কোরে (Core) একটি পরিবর্তি চৌম্বক ক্ষেত্র (Alternating Magnetic Field) সৃষ্টি হয়। হেডের সামনের অংশ যেখানে ফিতের সঙ্গে ঠেকে থাকে, সেখানে কোরের মধ্যে অতি সূক্ষ্ম একটি ফাঁক থাকে। এক বলে হেড গ্যাপ (Head Gap)। কোরে আবিষ্ট (Induced) চৌম্বক ক্ষেত্র, ওই ফাঁকের কাছে কোর ছেড়ে বাইরে বেরিয়ে আসে। এর প্রভাবে হেডের সামনে দিয়ে সমান গতিতে চলমান চৌম্বক ফিতেতে শব্দ অনুযায়ী একটি পরিবর্তি চৌম্বক ক্ষেত্র সৃষ্টি হয়ে যায়। হেডের সামনে থেকে ফিতেটি সরে গেলেও তাতে ওই চৌম্বক ক্ষেত্র অবিকৃতই থাকে। ফলে ফিতের ওপর অদৃশ্য চৌম্বক ক্ষেত্রের সাহায্যে রেকর্ডিং করা শব্দের তথ্য লিখিত হয়ে যায়।

পুনরায় ওই শব্দ শুনতে হলে হেডের সামনে দিয়ে টেপকে একই গতিতে চালানো হয়। এতে হেড গ্যাপের সামনে একটি পরিবর্তি চৌম্বক ক্ষেত্র সৃষ্টি হয়। এই চৌম্বক ক্ষেত্রের প্রভাবে হেড কোরেও



চৌম্বক পদার্থে তথ্য সংরক্ষণ

চৌম্বক ক্ষেত্র আবিষ্ট হয় এবং কোরের গায়ে জড়ানো হেড কয়েলে একটি পরিবর্তিত বিদ্যুৎ স্পন্দন সৃষ্টি হয়। বিদ্যুৎ স্পন্দনটি যে শব্দ রেকর্ড করা হয়েছিল তার সঙ্গে প্রায় হুবহু সম আকৃতির হয়। ফলে হেড কয়েলে আবিষ্ট বিদ্যুৎ স্পন্দনকে বিবর্ধিত করে স্পিকার বাজালে, যে শব্দ রেকর্ড করা হয়েছিল, তাকে ফিরে শোনা যায়।

কিন্তু ডিজিটাল সিগন্যাল রেকর্ড করার পদ্ধতি একটু অন্যরকম। বস্তুত ডিজিটাল সিগন্যাল রেকর্ড করতে বহু ধরনের পদ্ধতি ব্যবহার করা হয়। তার মধ্যে জনপ্রিয় দু-তিনটি পদ্ধতি সম্বন্ধে খুব সংক্ষেপে আলোচনা করা হলো।

NRZ (নন্ রিটার্ন জিরো)

নন্ রিটার্ন জিরো (Non Return Zero) কথা থেকে NRZ কথাটির উৎপত্তি। এই ব্যবস্থায় বাইনারি 1 বোঝাতে টেপের ওপর কোন এক দিকে চৌম্বক ক্ষেত্র সৃষ্টি করা হয়। বাইনারি 0 কে বোঝাতে ঠিক তার উল্টো দিকে চৌম্বক ক্ষেত্র সৃষ্টি করা হয়। অর্থাৎ এক কথায় বাইনারি সিগন্যাল অনুসারে টেপের চৌম্বক ক্ষেত্রের দিক পরিবর্তন করা হয়।

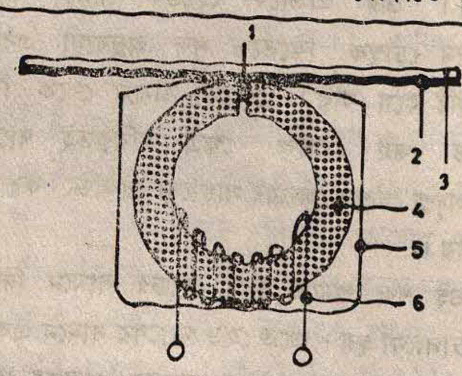
PHASE ENCODING

এই ব্যবস্থায় চৌম্বক ক্ষেত্রের পরিবর্তন ঘটে 0 থেকে 1 কিম্বা 1 থেকে 0 হবার মাঝামাঝি সময়ে।

PULSE WIDTH MODULATION

PWM ব্যবস্থা ডিজিটাল ক্যাসেট রেকর্ডারে বিশেষ ভাবে ব্যবহার করা হয়। এটির সাহায্যে যে কোনো সাধারণ গান টেপ করার টেপ রেকর্ডারে এবং ক্যাসেটে কম্পিউটারের তথ্য রেকর্ড করা যায়। পাল্‌স ওয়াইডথ মডিউলেশনে বাইনারি 1 এবং 0 বোঝাতে দু-রকম ডিউটি

চিত্র ৩২



রিড/রাইট (Read/Write) হেডের সাহায্যে চৌম্বক ফিতেতে পরিবর্তি চৌম্বক ক্ষেত্র সৃষ্টি করে তথ্য সংরক্ষণ পদ্ধতি

1. হেড গ্যাপ
2. ফেরো ম্যাগনেটিক পদার্থের প্রলেপ
3. মাইলারের তৈরী ফিতে
4. হেড কোর
5. ধাতব অবারণ 6. কয়েল

সাইকেল বিশিষ্ট একই কম্পাঙ্কের বিদ্যুৎ স্পন্দন হেড কয়েলের মধ্যে দিয়ে পাঠানো হয়। কিছু কিছু পাল্স ওয়াইডথ্ ব্যবস্থায় বাইনারি 1 বোঝাতে একটি নির্দিষ্ট কম্পাঙ্কের সাইন ওয়েভ টেপের : ওপর রেকর্ড করা হয়। বাইনারি 0-র সময়ে টেপের ওপর কোনো কম্পাঙ্কই রেকর্ড করা হয় না। এ ধরনের ব্যবস্থা টোন বার্স্ট (Tone Burst) পদ্ধতি হিসেবে বেশী পরিচিত।



ডিজিটাল ক্যাসেট রেকর্ডার

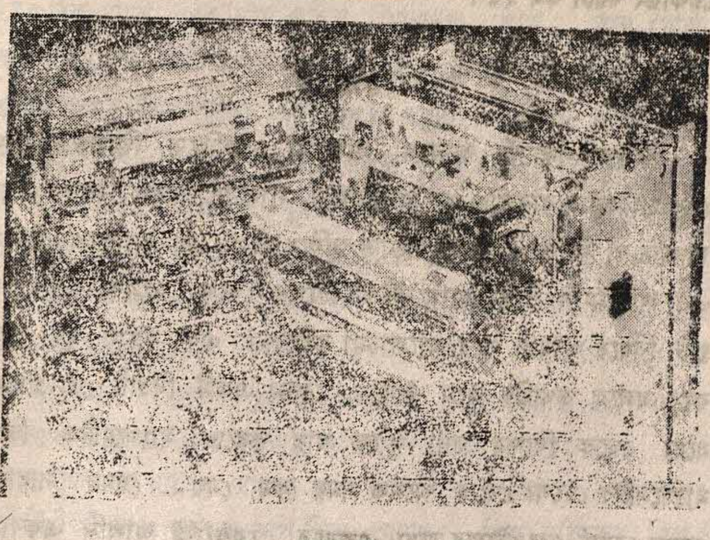
DOUBLE FREQUENCY

বাইনারি 1 বোঝাতে এই ব্যবস্থায় টেপের ওপর দুটি চৌম্বক স্পন্দন রেকর্ড করা হয়। 0 বোঝাতে রেকর্ডিং এর সময়ে কেবল একটি স্পন্দন রেকর্ড করা হয়।

ডিজিটাল ক্যাসেট রেকর্ডার

মাইক্রো কম্পিউটারের বাড়তি স্মৃতি হিসেবে ডিজিটাল ক্যাসেট রেকর্ডার খুবই জনপ্রিয়। সাধারণ ক্যাসেট রেকর্ডারে টোন বার্স্ট বা পালস্ ওয়াইডথ মডিউলেশন এর সার্কিট জুড়ে দিয়ে অনায়াসে বাইনারি তথ্য রেকর্ড করা যায়। কিছু কিছু ডিজিটাল ক্যাসেট রেকর্ডার অবশ্য এই কাজের জন্য বিশেষ ভাবে তৈরী করা অবস্থাতেই কিনতে পাওয়া যায়। এ সব ক্যাসেট রেকর্ডারে সাধারণ গান বাজাবার ক্যাসেট লাগানো হলেও এদের টেপ ট্রান্সপোর্ট ব্যবস্থা আরো উন্নত হয় এবং পালস্ ওয়াইডথ্ ছাড়াও ফেজ্ এনকোডিং কিম্বা ক্যান্সাস সিটি স্টানডার্ড পদ্ধতিতে ডিজিটাল তথ্য রেকর্ড করা হয়।

চিত্র ৩৩



ডিজিটাল ক্যাসেট রেকর্ডারের সাহায্যে সাধারণ গান টেপ করার ক্যাসেটে কম্পিউটার প্রোগ্রাম সংরক্ষণ করা হচ্ছে।

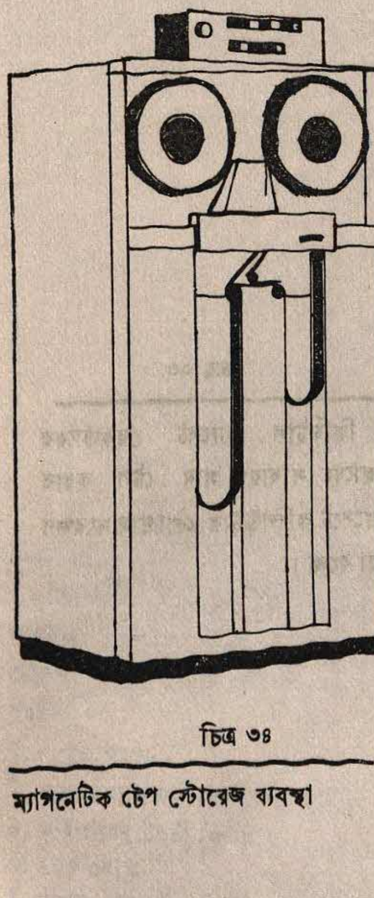


চৌম্বক স্মৃতি সংরক্ষণ ব্যবস্থা

সমস্ত চৌম্বক স্মৃতির মধ্যে এরাই হোলো সব থেকে কম দামী এবং সব থেকে কম শক্তিশালী স্মৃতি ব্যবস্থা। বড় কম্পিউটারে, যেমন মিনি বা মেইন ফ্রেমে, ডিজিটাল ক্যাসেট রেকর্ডার বিশেষ ব্যবহার করা হয় না। তবে ছোট মাইক্রো কম্পিউটারে এ ধরনের বাড়তি স্মৃতি অত্যন্ত জনপ্রিয়।

ম্যাগনেটিক টেপ স্টোরেজ

বড় বড় কম্প্যিউটারে ডিজিটাল তথ্য রেকর্ড করার জন্য বিশেষ ম্যাগনেটিক টেপ রেকর্ডার ব্যবহার করা হয়। এর জন্য যে ফিতে ব্যবহার করা হয় সেটিও হয় অনেক চওড়া, সর্বাধিক 3 ইঞ্চি। এই ফিতেতে একই সঙ্গে আটটি ষোলোটি বা বত্রিশটি ট্র্যাকে পুরো আট, ষোলো বা বত্রিশ বিট্ তথ্য রেকর্ড করা যায়। এই সুবিধার জন্য সাধারণ ডিজিটাল ক্যাসেটের থেকে এ ধরনের ম্যাগনেটিক মেমরির ডেটা অ্যাকশেস সময় অনেক কম লাগে, তাছাড়া এ ধরনের রেকর্ডারে তথ্য রেকডিং করা হয় অত্যন্ত দ্রুত গতিতে। রিডরাইট হেডের সামনে দিয়ে ম্যাগনেটিক ফিতেটি চালানো হয় সেকেণ্ডে 50 ইঞ্চি থেকে 200 ইঞ্চি গতিতে। মিনি আর মেইন ফ্রেম কম্পিউটারে এ ধরনের স্মৃতি ব্যবস্থা খুব বেশী ব্যবহার করা হয়। রেকর্ডার যন্ত্রটি বেশ দামী হলেও টেপের দাম কম হওয়াতে প্রতি বিট্ তথ্য রেকডিং খরচা কম হয়।

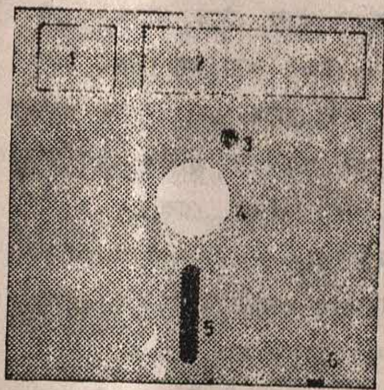


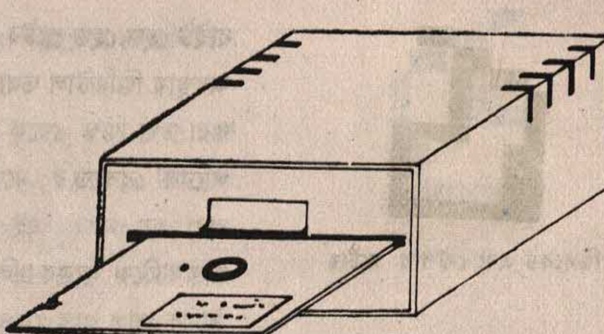
চিত্র ৩৪

ম্যাগনেটিক টেপ স্টোরেজ ব্যবস্থা

ডিসকেট

চৌম্বক পদার্থের আবরণ লাগানো চাকতিতে ডিজিটাল তথ্য রেকডিং করার ব্যবস্থাও খুব জনপ্রিয়। সাধারণ চৌম্বক ফিতের থেকে চাকতি বা ডিস্কের (Disc) বাড়তি সুবিধে হলো, এর অ্যাকশেস টাইম, এবং ল্যাটেনসি টাইম ম্যাগনেটিক টেপের থেকে অনেক কম। এ দুটি বিষয়ে আমরা পরে আলোচনা করবো। এক কথায় খুব সহজ করে বললে কথাটির মনে দাঁড়ায়, কোন একটি বিরাট প্রোগ্রামের মধ্যে থেকে বিশেষ তথ্য খুঁজে বার করতে ম্যাগনেটিক ডিস্ক, ম্যাগনেটিক টেপের থেকে অনেক সময় কম নেয়। যেমন সাধারণ গানের রেকর্ড এবং টেপের মধ্যে একমাত্র রেকর্ডেই আপনি একবারে





চিত্র ৩৫

কোন একটি বিশেষ গান খুঁজে বার করে চালাতে পারেন, ঠিক সে রকম ব্যাপার আর কি !

ম্যাগনেটিক ডিস্ক সংরক্ষণ ব্যবস্থায় দুধরণের যন্ত্র খুব জনপ্রিয়। সে দুটি হলো ডিসকেট (ফ্লপি ডিস্ক) আর হার্ড ডিস্ক। ফ্লপি ডিস্ক উদ্ভাবন করেন IBM কোম্পানী। এগুলি দেখতে হয় ইপি রেকর্ডের খাপের মতো। একটি 8 ইঞ্চি × 8 ইঞ্চি প্লাস্টিকের খামের মধ্যে 7·8" ইঞ্চি ব্যাসের মাইলারের তৈরী চাকতি ভরা থাকে। চাকতিটি অত্যন্ত পাতলা (এক ইঞ্চির কয়েক হাজার ভাগের মতো), তাই এর নাম দেওয়া হয়েছে ফ্লপি (Floppy)। চাকতিটির ওপর ম্যাগনেটিক টেপের মতই আয়রণ অক্সাইডের প্রলেপ দেওয়া হয়। যে খামের মধ্যে চাকতিটি ঢোকানো থাকে, তাতে তিনটি ফুটো থাকে। একটি বড় গোল ফুটো থাকে খামের ঠিক মাঝে, এর নাম ড্রাইভ স্পিনড্‌ল হোল। তার থেকে একটু দূরে ছোট একটি গোল ফুটো, এর নাম ইনডেক্স হোল, আর এছাড়া খামের ওপর লম্বাটে একটি ফুটো থাকে, এর নাম হেড স্লট। খামটি কোম্পানী থেকেই সিল করা থাকে যাতে ডিস্কটি না বার করা যায়। খামের অভ্যন্তরেই ঘুরন্ত চাকতি আর খামের মধ্যে ঘর্ষণ কমানোর জন্য বিশেষ dry lubrication-এর ব্যবস্থা থাকে।

খামটিকে অর্থাৎ ফ্লপি ডিস্ককে ফ্লপি ডিস্ক ড্রাইভ রেকর্ডারে ঢুকিয়ে দিলে রেকর্ডারের মটোর স্পিনডল, খামের ড্রাইভ স্পিনডল হোলের মধ্যে ঢুকে যায়। তারপর দুদিক থেকে দুটি গোল চাকতি এসে খামের স্পিনডল হোলের ভিতর বেরিয়ে আসা ফ্লপি ডিস্ককে চেপে ধরে। ডিস্ক ড্রাইভ চালালেই মটোর খামের মধ্যে ডিস্কটাকে ঘোরায়। সাধারণত মিনিটে 360 পাক গতিতে ফ্লপি ডিস্ক ঘোরে। ডিস্ক ড্রাইভের রিড্/

(ক আধুনিক ডিসকেট (Diskett) বা ফ্লপি ডিস্ক (Floppy Disc)

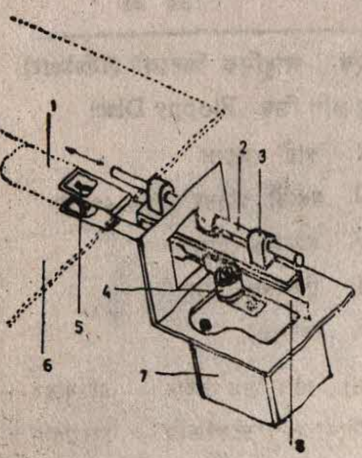
1. স্থায়ী লেবেল
2. অস্থায়ী পরিচয় জ্ঞাপক লেবেল
3. ইনডেক্স হোল
4. স্পিনডল্ হোল
5. হেড স্লট

(খ) ফ্লপি ডিস্ক ড্রাইভ। এই যন্ত্রের সাহায্যে ফ্লপি ডিস্কে ডিজিটাল তথ্য লেখা হয় কিম্বা লিখিত তথ্য পড়া হয়।



ডিসকেট এবং স্টেপার মটোর

রাইট হেড, হেড স্লটের মধ্যে দিয়ে ফ্লপির গায়ে ঠেকে থাকে। এই অবস্থায় ডিজিটাল তথ্য ডিস্কের ওপর রেকর্ড করা হয় কিম্বা রেকর্ড করা তথ্য ডিস্ক থেকে পড়া হয়। তবে ডিস্কে তথ্য লেখার সময় গানের রেকর্ডের মতো প্যাঁচালো বা স্পাইরাল ট্র্যাকে রেকর্ডিং করা হয় না। এই কাজের জন্য ফ্লপি ডিস্কের রেকর্ডিং করার জায়গাটিকে অনেকগুলি সমকেন্দ্রিক বলয় বা Concentric annular ট্র্যাকে ভাগ করে নেওয়া হয়। সাধারণত একটি আট ইঞ্চি ফ্লপির প্রতি দিক 76টি ট্র্যাকে ভাগ করা হয়। ট্র্যাকে ভাগ করে তথ্য লেখার দরুণ ফ্লপি ডিস্কের রিড/রাইট হেডকে কোন একটি ট্র্যাকে চট্ করে নিয়ে আসা খুব সুবিধাজনক হয়। এই কাজের জন্য স্টেপার মটোর ব্যবহার করা হয়। স্টেপার মটোরের বৈশিষ্ট্য হলো, এতে একাধিক (সাধারণত চারটি) স্টেটর কয়েল (Stator Coil) থাকে। যে কোনো স্টেটর কয়েলে একটি বিদ্যুৎ স্পন্দন পাঠালে এটি সাধারণ মটোরের মতো ক্রমাগত না ঘুরে একটি নির্দিষ্ট ডিগ্রি পর্যন্ত পাক খেয়ে থেমে যায়।



চিত্র ৩৬

পুনরায় তাকে ঘোরাতে হলে পূর্বের স্টেটর কয়েলের বিদ্যুৎ প্রবাহকে বন্ধ করে তার ঠিক পরের কয়েলে বিদ্যুৎ প্রবাহ পাঠাতে হয়। এতে সেটি আবার সেই নির্দিষ্ট ডিগ্রি পর্যন্ত ঘুরে থেমে যায়। এইভাবে পর্যায়ক্রমে সমস্ত স্টেটর কয়েলে একবার করে একটি বিদ্যুৎ স্পন্দন পাঠিয়ে স্টেপার মটোরকে ঘোরানো হয়। বিদ্যুৎ প্রবাহের দিক পাল্টালে স্টেপার মটোর উল্টোদিকে ঘোরে। এই ব্যবস্থার সাহায্যে খুব সহজেই রিড/রাইট হেডকে বিভিন্ন ট্র্যাকে আনা যায়। ধরা যাক হেডটি প্রথম ট্র্যাকে আছে। তাকে দশম ট্র্যাকে আনতে হলে হেড

---

1. হেড স্লট।
2. ক্যারেজ ওয়ে।
3. ক্যারেজ গাইড।
4. স্টেপার মোটরের ক্যাপস্টান।
5. রিড/রাইট হেড।
6. ডিসকেট।
7. স্টেপার মোটর।
8. ধাতব ব্যাণ্ড।

স্টেপার মোটরের সাহায্যে ফ্লপি ডিস্কের রিড/রাইটকে বিভিন্ন ট্র্যাকে নিয়ে যাওয়ার ব্যবস্থা। ট্র্যাক পরিবর্তনের দরকার হলে, স্টেপার মোটরটির ক্যাপস্টান (4) ঘোরে। ক্যাপস্টানের গায়ে লাগানো ধাতব ব্যাণ্ড (8) এর ফলে এগিয়ে বা পিছিয়ে যায়। ব্যাণ্ডের সঙ্গে সংযুক্ত সম্পূর্ণ রিড রাইট ক্যারেজটিও সেই সঙ্গে ডিসকেটের হেড স্লটে স্থান পরিবর্তন করে।

পজিশনিং স্টেপার মটোরে দশটি বিদ্যুৎ স্পন্দন পাঠালেই হেডটি দশম ট্র্যাকে গিয়ে দাঁড়িয়ে যাবে।

ইনডেক্স হোলের কাজ হলো, কখন কোন ট্র্যাকে তথ্য লেখা শুরু হলো সেটি জানান দেওয়া। এর জন্য খামের ওপর যেখানে ইনডেক্স হোল আছে, ফ্লপিতেও ওই একই জায়গায় একটি ফুটো করা থাকে। প্রতিবার ডিস্কটি খামের মধ্যে ঘোরার সময় একবার করে খামের এবং চাকতির ফুটো একসঙ্গে মিলিত হয়। সেই সময়ে ফ্লপির এক দিকে রাখা একটি আলোক উৎস থেকে বিচ্ছুরিত আলো ওই ফুটো দিয়ে উল্টো দিকে রাখা ফোটো ডিটেক্টর সার্কিটে পড়ে, আর বিদ্যুৎ স্পন্দন সৃষ্টি করে। এই বিদ্যুৎ স্পন্দন থেকে একটি বিশেষ ইলেকট্রনিক সার্কিট, ফ্লপি ড্রাইভ নিয়ন্ত্রণ ব্যবস্থাকে জানান দেয় এই জায়গা থেকে সমস্ত ট্র্যাকগুলি শুরু হচ্ছে।

এছাড়া ফ্লপি ডিস্কে, ভুলবশত রেকর্ডিং না হবার ব্যবস্থাও থাকে। এটিকে বলা হয় রাইট প্রোটেকশন নচ্। নতুন ফ্লপিতে ওই জায়গাটি অটুট থাকে। কোনো তথ্য রেকর্ড করার পর ওই নচ্ ভেঙে বা ছিঁড়ে দিলে সেই ফ্লপিতে নতুন তথ্য আর রেকর্ড করা যায় না।

বর্তমানে 8 ইঞ্চি ফ্লপি বড় বড় কম্পিউটারে ব্যবহার করা হয়। ছোট বা মিনি ফ্লপি, যেগুলির সাইজ হয় 5·25 ইঞ্চি এবং 3·5 ইঞ্চি, সেগুলি মাইক্রো কম্পিউটারে খুব বেশী ব্যবহার করা হয়। ফ্লপি এখন খুবই জনপ্রিয় স্টোরেজ ব্যবস্থা। সমস্ত পারশোনাল কম্পিউটারে এবং মাইক্রো কম্পিউটারে বাড়তি স্মৃতি হিসেবে প্রায় সব সময়ই ফ্লপি ব্যবহার করা হয়।

হার্ড ডিস্ক ড্রাইভ

নমনীয় পদার্থের ডিস্ক ছাড়া আর এক ধরনের ম্যাগনেটিক ডিস্ক স্টোরেজ যন্ত্র আছে। এই যন্ত্রে শক্ত ডিস্ক (Hard Disc) ব্যবহার করা হয়, তাই এদের নাম হার্ড ডিস্ক ড্রাইভ। কম্পিউটার জগতে। উইনচেস্টার হার্ড ডিস্ক ড্রাইভ (Winchester Hard Disc Drive) নামেও এরা পরিচিত। ফ্লপির সঙ্গে উইনচেস্টারের গঠন এবং কার্যগত অনেক পার্থক্য আছে।



হার্ড ডিস্ক

ফ্লপি ডিস্ক নাড়াচাড়া করার বিষয়ে বিশেষ সাবধানতা

১। ব্যবহার না করা অবস্থায় সবসময় ফ্লপিকে তার বাক্সে পুরে রাখতে হয়।

২। হেড স্লটে কোন অবস্থাতেই আঙুল ছোঁয়ানো চলবে না।

৩। ফ্লপিকে স্পিকার, টি ভি, ট্রান্সফরমার বা অন্য কোন রকম চৌম্বক ক্ষেত্রের কাছাকাছি রাখা চলবে না। যে কোন রকম শক্তিশালী চৌম্বক ক্ষেত্র বা উচ্চ তাপমাত্রা ফ্লপিতে রেকর্ডিং করা তথ্য মুছে ফেলতে পারে।

৪। ফ্লপি লাগানো অবস্থায় ডিস্ক ড্রাইভ বন্ধ করা অনুচিত। কিছু কিছু ডিস্ক ড্রাইভ বন্ধ হওয়ার সময় একটি শক্তিশালী চৌম্বক ক্ষেত্রের সৃষ্টি করে, ফ্লপিতে সংরক্ষিত তথ্যকে মুছে দিতে পারে।

৫। ফ্লপির গায়ে কিছু লিখতে হলে অত্যন্ত হাল্কা ভাবে কোনো চাপ প্রয়োগ না করে লিখতে হয়।



হার্ড ডিস্ক

১। উনচেস্টারে শক্ত ডিস্ক ব্যবহার করা হয়।

২। এই ডিস্কগুলি ড্রাইভের মধ্যে চিরস্থায়ী করে বসানো, ফ্লপির মতো ইচ্ছে মতো পাল্টানো যায় না।

৩। এতে এক সঙ্গে একাধিক ডিস্ক পরপর রেখে রিড/রাইট কাজ চালানো হয়।

৪। ফ্লপির মতো রিড রাইট হেডটি ডিস্কের সঙ্গে ঠেকে থাকে না। ডিস্কের ওপর পাতলা বাতাসের স্তরের ওপর ভেসে থাকে।

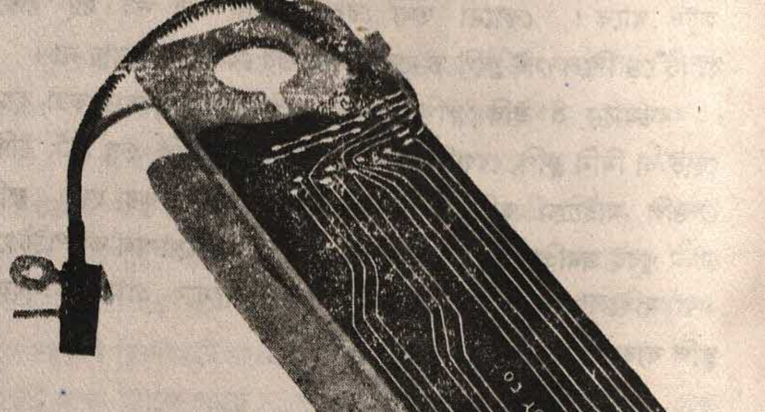
৫। উইনচেস্টার ডিস্ক ফ্লপির থেকে আরো জোরে ঘোরে (মিনিটে 1000 পাক থেকে 3600 পাক পর্যন্ত)।

৬। ফ্লপির থেকে এতে আরো বেশী তথ্য রেকর্ড করা যায়।

একটি সম্পূর্ণ ধুলো বিহীন প্লাস্টিকের চেম্বারে হার্ড ডিস্কগুলি পরপর সাজানো থাকে। প্রতি ডিস্কেটির ওপরে এবং নিচে দুদিকেই তথ্য রেকর্ড করার জন্য রিড/রাইট হেড রাখা হয়। এই হেডগুলি স্টেপার

চিত্র ৩৭

উইনচেস্টার হার্ড ডিস্ক ড্রাইভে ব্যবহৃত ভাসমান রিড/রাইট হেড (Flying Read/Write Read)

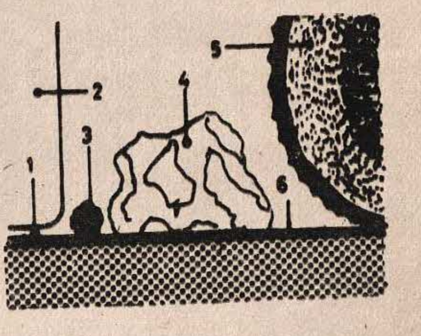


মোটরের সাহায্যে বিভিন্ন ট্র্যাকে নিয়ে আসা যায়। ডিস্কগুলি 1000 থেকে 3600 পাক প্রতি মিনিটে ঘুরতে শুরু করলে, ডিস্কের ঠিক গায়ে লেগে থাকা বাতাসের স্তরও ওই একই গতিতে ডিস্কের সঙ্গে ঘুরতে থাকে। এ অবস্থায় হেডগুলিকে স্প্রিং-এর সাহায্যে ঘূর্ণায়মান ডিস্কের ওপর চেপে ধরা হয়। কিন্তু ওই দ্রুত গতিতে ডিস্কের সঙ্গে ঘুরতে থাকা বাতাসের স্তর তখন শক্ত পদার্থের মত আচরণ করে। বাতাসের স্তরটি এতো শক্ত হয়ে যায় যে তার স্টিফনেস (Stiffness) কয়েক হাজার পাউণ্ড প্রতি স্কোয়ার ইঞ্চিকেও ছাড়িয়ে যায়! অর্থাৎ হেডগুলিকে স্প্রিংয়ের সাহায্যে বেশ জোরে ডিস্কের সঙ্গে চেপে ধরার চেষ্টা করলেও কিছুতেই ডিস্কের সঙ্গে লাগানো যায় না। একটি অত্যন্ত পাতলা (এক ইঞ্চির এক লাখ ভাগের এক ভাগ) বাতাসের স্তরের ওপর সেটি ভেসে থাকে। ডিস্কের সঙ্গে হেডের সরাসরি কোন সংযোগ না থাকায় ডিস্ক এবং হেডের মধ্যে কোন ঘর্ষণজনিত ক্ষয় হয় না। এটিই উইনচেস্টার ডিস্কের অন্যতম সুবিধে। তবে ডিস্ক থেকে অতো অল্প দূরত্বে হেডটি ভেসে থাকার জন্য ঘুরন্ত ডিস্কের চেম্বারটা সম্পূর্ণ ধুলো বিহীন হতে হয়। 10 মাইক্রো ইঞ্চি দূরত্বটি এতোই কম যে তার তুলনায় সিগারেটের ধোঁয়ায় ভেসে থাকা কার্বন কণিকাও বিশাল পাহাড়ের মতো বড় দেখায়। ফলে ঘুরন্ত ডিস্কের চেম্বারে যদি সামান্য ধোঁয়াও প্রবেশ করে তবে সেটির কার্বন কণিকা ডিস্ক আর হেডের মাঝে ঢুকে পড়ে বিরাট অনর্থ বাঁধিয়ে দেয়। সেইজন্য উইনচেস্টার ড্রাইভের সমস্ত ইলেকট্রনিক যন্ত্রপাতি এবং মোটরকে ঠাণ্ডা রাখার জন্য পাখা এবং বায়ু ধুলো নিরুদ্ধে ভাবে কোম্পানি থেকে সিল করে দেওয়া হয়। যন্ত্রের ভেতরে যে সম্পূর্ণ পরিষ্কার ধুলো মুক্ত বাতাস থাকে, তাকেই ব্লোয়ারের সাহায্যে মোটর এবং সার্কিটের ওপর দিয়ে প্রবাহিত করে যন্ত্রটিকে ঠাণ্ডা রাখা হয়।

এ ধরনের স্মৃতি, ফ্লপির থেকে অনেক বেশী দামী, এবং দ্রুত। তাই উইনচেস্টার হার্ড ডিস্ক ড্রাইভ মিনি আর মেইন ফ্রেমে বেশী ব্যবহার করা হয়। অবশ্য বেশী দামী মাইক্রো কম্পিউটারেও অনেক সময়ে উইনচেস্টার হার্ড ডিস্ক ব্যবহার করা হয়ে থাকে।



হার্ড ডিস্ক



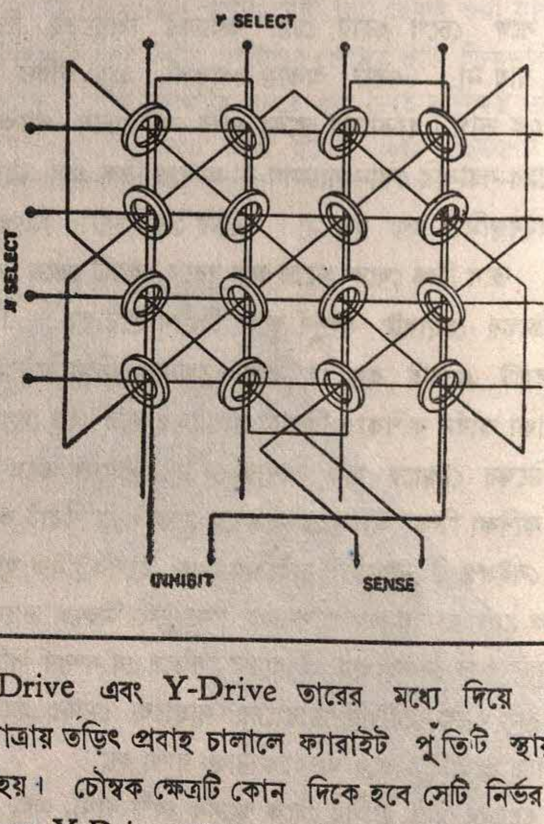
চিত্র ৩৮

1. ঘুরন্ত হার্ড ডিস্ক এবং ভাসমান রিড/রাইট হেডের মধ্যে 10 মাইক্রণ ফাঁক।
2. রিড/রাইট হেডের আংশিক প্রস্থচ্ছেদ
3. সিগারেটের ধোঁয়ার কার্বন কণা
4. বাতাসে ভাসমান সূক্ষ্ম ধুলো
5. মানুষের মাথার চুলের প্রস্থচ্ছেদ
6. হার্ড ডিস্কের চৌম্বক প্রলেপ

ম্যাগনেটিক কোর স্মৃতি ব্যবস্থা

ম্যাগনেটিক কোর স্মৃতি

ম্যাগনেটিক কোর স্মৃতিতে বাইনারি তথ্য ধরে রাখতে খুব ছোট ছোট ফ্যারাইটের তৈরী পুঁতির (Bead) চৌম্বক ক্ষেত্রের সাহায্য নেওয়া হয়। 1·27 মিলিমিটার ব্যাসের পুঁতিগুলোর মধ্যে ·76 মিলিমিটার ব্যাসের সূক্ষ্ম ফুটো করা থাকে এবং সেই ফুটোয় চারটি সরু সরু তার গলানো থাকে। এই চারটি তারের নাম যথাক্রমে X-Drive Y-Drive Sense এবং Inhibit তার।



চিত্র ৩৯

ম্যাগনেটিক কোর তথ্য সংরক্ষণ ব্যবস্থা

X-Drive এবং Y-Drive তারের মধ্যে দিয়ে এক সঙ্গে নির্দিষ্ট মাত্রায় তড়িৎ প্রবাহ চালালে ফ্যারাইট পুঁতিটি স্থায়ী চুম্বকে পরিণত হয়। চৌম্বক ক্ষেত্রটি কোন দিকে হবে সেটি নির্ভর করে X-Drive এবং Y-Drive তারের মধ্যে দিয়ে প্রবাহিত বিদ্যুৎ প্রবাহের দিকের ওপর। সাধারণত দক্ষিণাবর্তি (Clockwise) চুম্বক ক্ষেত্রের সাহায্যে বাইনারি 1 এবং বামাবর্তি (Anticlokwise) ক্ষেত্রের সাহায্যে বাইনারি 0 তথ্য লিপিবদ্ধ করা হয়। এরপর X, Y ড্রাইভে বিদ্যুৎ প্রবাহ বন্ধ করে দিলেও পুঁতির চুম্বক ক্ষেত্রে বাইনারি তথ্য লিপিবদ্ধ থেকে যায়। পুঁতিকে চুম্বক করতে X, Y ড্রাইভ দুটো তার ব্যবহারের কারণ, ম্যাগনেটিক কোর মেমরিতে কোরগুলিকে ম্যাট্রিক্স

ছকে সাজানো হয় এবং ওই ম্যাট্রিক্স ছকের মধ্যে অবস্থিত কোন একটি বিশেষ কোরকে বেছে নিয়ে তাতে তথ্য লেখা বা পড়ার কাজে ও দুটি তার কাজে লাগে (ঠিক যেমন ইলেক্ট্রনিক RAM স্মৃতিতে করা হয়)।

ওই ফ্যারাইট কোর বা পুঁতি থেকে তথ্য পড়ার সময় X এবং Y ড্রাইভ তারের মধ্যে দিয়ে বাইনারি 0 লেখার বিদ্যুৎ প্রবাহ পাঠানো হয়। পুঁতিতে যদি আগে থেকে বাইনারি 1 তথ্য লেখা থাকে তবে X, Y ড্রাইভের প্রবাহের প্রভাবে পুঁতির চৌম্বক ক্ষেত্রের দিক বাইনারি 1 থেকে মুহূর্তের মধ্যে বাইনারি 0 -এ পরিবর্তিত হয়। হঠাৎ চৌম্বক ক্ষেত্রের পরিবর্তনে Sense wire-এ একটি বৈদ্যুতিক স্পন্দন আবিষ্ট হয়। এই স্পন্দন থেকেই C.P.U. বুঝতে পারে ওই মেমরি লোকশনে বাইনারি 1 তথ্য লেখা ছিলো।

যদি পুঁতিতে আগে থেকে বাইনারি 0 তথ্য লিপিবদ্ধ থাকে তবে X, Y এর প্রবাহের ফলে পুঁতির চৌম্বক ক্ষেত্রে কোনোই পরিবর্তন ঘটেনা। ফলে Sense তারেও কোনো স্পন্দন আবিষ্ট হয় না। এর থেকে C.P.U বুঝতে পারে ওই মেমরি লোকেশনে 0 তথ্য লিপিবদ্ধ আছে।

এ পর্যন্ত সব কিছুই বেশ নির্ঝঞ্ঝাট। কিন্তু একটি ব্যাপার লক্ষ্য করুন, তথ্য পড়ার পরে প্রতিটি ফ্যারাইট কোরেই কিন্তু বাইনারী 0 তথ্য লিপিবদ্ধ হয়ে যায় (DRAM স্মৃতির মতো)। সুতরাং যা কিছু তথ্য স্মৃতিতে লেখা ছিলো, সমস্তই একেবারে মুছে যায়। এই ঝামেলা এড়াতে প্রতিটি কোরের তথ্য পড়া হলেই সমস্ত X, Y ড্রাইভে বাইনারি 1 লেখার বিদ্যুৎ প্রবাহ পাঠানো হয়। একই সঙ্গে যে সব সেলে পূর্বে বাইনারি 0 তথ্য লেখা ছিলো, শুধু তাদের Inhibit তারে বাইনারি 0 লেখার বিদ্যুৎ প্রবাহ পাঠানো হয়। Inhibit তারের প্রবাহের প্রভাবে X, Y ড্রাইভ ওইসব কোরে বাইনারি 1লিখতে পারেনা। তাই তারা পূর্বের মতই বাইনারি 0 তে থেকে যায়। আর যে সব কোরে বাইনারি 1 তথ্য লেখা ছিলো, তাদের Inhibit তারে কোনো প্রবাহ না থাকায় তাদের পুঁতিতে পুনরায় বাইনারি 1 তথ্য লেখা হয়ে যায়।

ম্যাগনেটিক কোর মেমরিতে লেখা তথ্য পড়লেই মেমরির তথ্য মুছে যায় বলে একে dynamic storage বলা যেতে পারে। তবে কম্পিউটার পরিভাষায় একে destructive storage



ম্যাগনেটিক কোর স্মৃতি



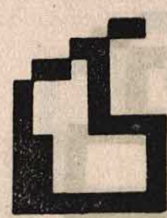
াবল মেমরি

বা ধ্বংসাত্মক স্মৃতি ব্যবস্থা বলে অভিহিত করা হয়। কোর স্মৃতিতে তথ্য লেখা এবং পড়ার জন্য প্রচুর কারেণ্ট দরকার পড়ে। তাছাড়া এদের আকৃতিও খুব ছোট করা যায় না। একটি 32 K বিট্ ম্যাগনেটিক কোর স্মৃতি দু-ইঞ্চি দৈর্ঘ্য, প্রস্থ, উচ্চতা বিশিষ্ট ঘনকের (Cube) জায়গা দখল করে নেয়। এ সবই হলো কোর স্মৃতির অসুবিধা। তবে উচ্চতাপমাত্রা এবং তীব্র ইলেক্ট্রোম্যাগনেটিক রেডিয়েশানে যখন অন্য সমস্ত স্মৃতি ব্যবস্থা অকেজো হয়ে পড়ে, তখন কোর স্মৃতি কিন্তু ঠিক ঠাক কাজ চালিয়ে যায়। এই কারণে কয়েকটি বিশেষ কাজে এবং কিছু কিছু মেইন ফ্রেম কম্পিউটারে এখনও ম্যাগনেটিক কোর স্মৃতি ব্যবহার করা হয়।

বাবল্ মেমরি

বাবল মেমরিতে বাইনারি তথ্য লিখে রাখার জন্য চৌম্বক বুদ্ বুদ্ বা Magnetic Bubble-এর সাহায্য নেওয়া হয়। এর প্রযুক্তি কৌশল এতো জটিল যে এ বিষয়ে বিশদ করে আলোচনা সম্ভব নয়। খুব সংক্ষেপে এবং অতি সরলিকৃতভাবে এদের কার্যপ্রণালী ব্যাখ্যা করা হলো।

একটি বাবল মেমরিতে গ্যাডোলিনিয়াম-গ্যালিয়াম-গার্ণেট (Gadolinium Gallium Garnet) বা সংক্ষেপে GGG-র পাতলা ওয়েফার বা সাবস্ট্রেটে অত্যন্ত পাতলা চৌম্বক গারনেটের স্তর তৈরী করা হয়। সাবস্ট্রেটের ওপরে এবং নিচে শক্তিশালী স্থায়ী চুম্বক রাখা হয় যাতে সেই চুম্বকের প্রভাবে গারনেটের প্রতিটি অনু বা Magnetic Domain একই দিকে মুখ করে থাকে। এই অনুগুলির মধ্যে যদি কোনটিকে জোর করে উল্টো দিকে ঘুরিয়ে দেওয়া যায় তবে যেটি সৃষ্টি হয়, তাকেই বলে ম্যাগনেটিক বাবল। গারনেটের মধ্যে এই ম্যাগনেটিক বাবল তৈরী করার জন্য সাবস্ট্রেটের ঠিক ওপরে অত্যন্ত সূক্ষ্ম নিকেল আয়রন কিম্বা পারমালয়ের তৈরী অজস্র ছুঁচের মতো তড়িৎ চুম্বক রাখা হয়। এই তড়িৎ চুম্বকের সাহায্যে গারনেটের ওপর প্রয়োজন মতো ম্যাগনেটিক বাবল তৈরী করা বা মুছে ফেলা যায়। চুম্বক, ছুঁচ এবং সাবস্ট্রেট, এই তিনটিকেই আবার দুটি অর্থোগোনাল (Orthogonal) পরিবাহী কয়েলের মধ্যে পুরে ফেলা হয়। অর্থোগোনাল কয়েল বলতে বোঝায় দুটি তারের কয়েল, যাদের চৌম্বক ক্ষেত্র দুটি পরস্পর সমকোনে অবস্থিত। অর্থোগোনাল কয়েলে বিশেষ



মাইক্রো ফিশ্

AC প্রবাহ পাঠিয়ে সাবস্ট্রেটের ওপর ঘূর্ণায়মান চৌম্বক ক্ষেত্র বা Rotating Magnatic Field তৈরী করা হয়। চৌম্বক ক্ষেত্রের প্রভাবে গার্ণেটের সমস্ত চৌম্বক অনুগুলি এবং তার মাঝে মাঝে যে সব বাবল্ আছে, সেগুলি চৌম্বক ছুঁচের তলা দিয়ে সাইকেলের চেনের মতো চক্রাকারে ঘুরে যেতে থাকে। বাবল্ মেমরিতে তথ্য লেখার সময়ে ছুঁচের সাহায্যে ঘূর্ণায়মান চৌম্বক অণুতে ইচ্ছেমত বাবল্ সৃষ্টি করা হয়, আর লিখিত তথ্য পড়ার জন্য ব্যবহার করা হয় বিশেষ বাবল্ ডিটেক্টর ব্যবস্থা। ডিটেক্টরের তলা দিয়ে যখন কোন বাবল্ যায় তখন ডিটেক্টরে কয়েক মিলি ভোল্ট বিদ্যুৎ স্পন্দন পাওয়া যায়। অর্থাৎ পুরো ব্যাপারটাই যেন ক্যাসেট টেপের মতো। হেডের সামনে দিয়ে যেমন ফিতেতে টেপ করা চৌম্বক ক্ষেত্র চলে যায়, বাবল্ মেমরিতে রোটেটিং ম্যাগনেটিক ফিল্ডের প্রভাবে ডিটেক্টরের তলা দিয়ে ম্যাগনেটিক ডোমেনগুলিও তেমনি সচল অবস্থায় থাকে। তবে ম্যাগনেটিক ডোমেন সচল থাকলেও টেপের ফিতের মতো বাবল্ মেমরিতে যান্ত্রিক ভাবে সচল কোনো যন্ত্রাংশ থাকে না। বর্তমানে পুরো জিনিসটি একটি আইসি-র প্যাকে পাওয়া যায়।

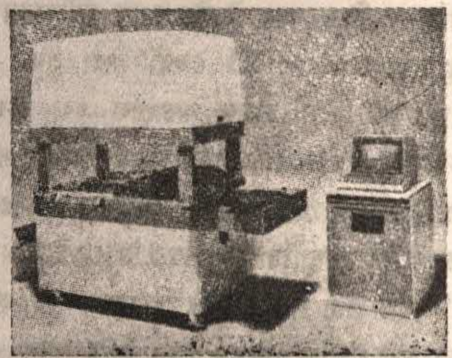
এই প্রযুক্তিতে বর্তমানে মাত্র 4·4 বর্গ সেন্টিমিটার সাবস্ট্রেটের ওপর ম্যাগনেটিক বাবল্ তৈরি করে এক লক্ষ বিট্ তথ্য ধরে রাখা সম্ভব হচ্ছে। ভবিষ্যতে এই পরিমাণ অবশ্যই আরো বাড়বে। তবে বাবল্ মেমরির প্রধান অসুবিধা হলো এটি ম্যাগনেটিক ডিস্কের মতো অতো দ্রুত কাজ করতে পারে না। এই অসুবিধাও ভবিষ্যতে নিশ্চয়ই দূর করা সম্ভব হবে।

মাইক্রোফিশ্

সমস্ত বাণিজ্যিক প্রতিষ্ঠানেই অজস্র নথী এবং হিসেবের কাগজ পত্র সযত্নে সংরক্ষণ করতে হয়। সাবেক কালের নিয়ম অনুসারে এতদিন এই সমস্ত নথী লেজার খাতা আর মোটা মোটা ফাইলের ধুলো ভর্তি স্তূপের মাধ্যমেই জমিয়ে রাখা হত। কিন্তু সর্বাধুনিক মাইক্রোফিশ্ (Microfiche) কম্পিউটার তথ্য সংরক্ষণ ব্যবস্থার সাহায্যে অনেক কম খরচে এবং বহুগুণ কম জায়গায় এই সমস্ত তথ্য সংরক্ষণ করা সম্ভব হচ্ছে।



মাইক্রো ফিশ্



চিত্র ৪০

ছবির বাঁদিকে দেখা যাচ্ছে মাইক্রোফিশ্ ছবি তোলা এবং প্রসেস করার স্বয়ংক্রিয় যন্ত্র। ডান দিকে রয়েছে কম্পিউটার ও মনিটর

মাইক্রোফিশ্ স্মৃতিভাণ্ডারের কর্মপদ্ধতি কিন্তু অন্য সমস্ত রকম কম্পিউটার স্মৃতিভাণ্ডারের থেকে আলাদা। এই ব্যবস্থায় যাবতীয় তথ্য সবই ফোটোগ্রাফিক নেগেটিভে আণুবীক্ষণিক হরফে লিখে রাখা হয়। সাধারণত এই কাজে 6 ইঞ্চি দৈর্ঘ্য এবং 4 ইঞ্চি প্রস্থ বিশিষ্ট নেগেটিভ ব্যবহার করা হয়। এই নেগেটিভকে বলা হয় ফিশ্ (Fiche)। Fiche শব্দটি ফরাসী, এবং এটি বলতে বোঝায় কার্ড (Card)। একটি ফিশ্ এ আড়াইশো পাতার বইয়ের সমস্ত শব্দ ধরে যায়। কিছু কিছু আল্টাফিশ্ (Ultrafiche) যন্ত্র ওই 6 × 4 কার্ডে হাজার পাতার তথ্যও লিখতে পারে।

কম্পিউটার থেকে আগত সমস্ত তথ্যকে প্রথমে মাইক্রোফিশ্ যন্ত্রের অভ্যন্তরে লাগানো V. D. U স্ক্রিনে ফুটিয়ে তোলা হয়। স্ক্রিনের দৃশ্যকে তারপর দ্রুত ছবি তুলতে সক্ষম ক্যামেরার সাহায্যে Fichefilm এ তুলে নেওয়া হয়। ক্যামেরাটি সাধারণত মিনিটে 32000 লাইনের ছবি তুলতে পারে। এরপর ফিল্মটি স্বয়ংক্রিয় ডেভলপিং এবং ফিক্সিং ব্যবস্থার মাধ্যমে প্রসেস হয়ে বেরিয়ে আসে। এই ভাবে তথ্য লিখে রাখতে লাইন প্রিন্টারের থেকেও কম সময় লাগে।

স্মৃতি ব্যবস্থা সংক্রান্ত কয়েকটি দরকারি তথ্য

আমরা ইতিপূর্বে বেশ কয়েকটি কম্পিউটার পরিভাষার সঙ্গে পরিচিত হয়েছে যেমন অ্যাকশেস টাইম, ল্যাটেনসি, র‍্যানডম অ্যাকশেস, সিরিয়াল অ্যাকশেস ইত্যাদি। এই বিষয়গুলি যেকোনো কম্পিউটার ব্যবহারকারীর পক্ষেই গুরুত্বপূর্ণ। খুব সংক্ষেপে সেই সব পরিভাষা ব্যাখ্যা করা হল।

অ্যাকশেস টাইম
ল্যাটেনসি

অ্যাক্শেস টাইম

স্মৃতি ব্যবস্থায় অ্যাকশেস টাইম (Access time) হল সেই স্মৃতি ব্যবস্থা থেকে কোন একটি বিশেষ তথ্য খুঁজে বের করে সেই তথ্য পড়তে যে সময় লাগে সেই সময়। অ্যাক্শেস টাইম যত কম হয় ততই ভালো। খুব দ্রুত কাজ কর্মের জন্য দ্রুত অ্যাক্শেস টাইম বিশিষ্ট স্মৃতি ব্যবস্থা প্রয়োজন। হার্ড ডিস্কের অ্যাক্শেস টাইম খুবই কম। আর RAM এর অ্যাক্শেস টাইম হলো সর্বনিম্ন।

ল্যাটেনসি

হার্ড ডিস্ক, ফ্লপি বা বাবল্, মেমরি জাতীয় স্মৃতিতে অ্যাক্শেস টাইম ছাড়াও ল্যাটেনসি (Latency) বলে একটি দেরী হয়। যে সমস্ত স্মৃতিতে তথ্যকে একটি আবর্তের মধ্যে লেখা হয়, সেখানেই ল্যাটেনর্সি ব্যাপারটি চলে আসে। যেমন, হার্ড ডিস্কে তথ্য লেখা হয় সমকেন্দ্রিক চক্রাকার ট্র্যাকে। ধরা যাক আমাদের দশ নম্বর ট্র্যাকের তথ্য পড়তে হবে। এই নির্দেশ দিলেই হার্ড ডিস্ক ড্রাইভের স্টেপার মটোর হেডকে দশ নম্বর ট্র্যাকে নিয়ে আসবে। হেডটি পূর্বে যে ট্র্যাকে ছিলো, সেখান থেকে দশ নম্বর ট্র্যাকে আসতে যে সময় নেবে সেটি অ্যাক্শেস টাইমের মধ্যে ফেলা হবে। কিন্তু দশ নম্বর ট্র্যাকে আসা মাত্রই হেডটি সেখানে লিখিত তথ্য পড়তে পারবেনা কারণ যে সময় হেডটি দশ নম্বরে আসবে ঠিক সেই সময়ই দশ নম্বর ট্র্যাকে লিখিত প্রথম তথ্যটি তার সামনে দিয়ে যাবেনা। তাই সঠিক ট্র্যাকে আসার পরও হেডটিকে অপেক্ষা করে থাকতে হবে কতক্ষণে সেই ট্র্যাকের প্রথম তথ্যটি তার নিচ দিয়ে যাবে। এই অপেক্ষার সময়টিকে বলা হয় ল্যাটেনসি। বাবল, মেমরিতে ল্যাটেনসির দরুন তথ্য পড়তে বেশ দেরী হয়, 400 থেকে 500 মিলি সেকেণ্ড মতো (একটি বাবল্, মেমরিতে এক মেগাবিট্ তথ্য লিপি বদ্ধ করা যায় আর এর মধ্যে কোন একটি তথ্য পড়তে গেলে সর্বাধিক এক মেগা বিট তথ্য ডিটেকটরের তলা দিয়ে যেতে যে সময় লাগে তত সময় আমাদের অপেক্ষা করতে হয়)। তবে একবার প্রয়োজনীয় তথ্য এসে গেলে পরবর্তী তথ্য পড়বার সময় বা অ্যাক্শেস টাইম খুবই কম লাগে।

র‍্যানডম অ্যাক্শেস
ডাইরেক্ট অ্যাক্শেস
সিরিয়াল
অ্যাক্শেস

র‍্যানডম অ্যাক্শেস

যে সব স্মৃতি ভাণ্ডারে যেকোনো মেমরি লোকেশন থেকে যখন খুশি তথ্য পড়তে চাওয়া যায় এবং পড়তে চাওয়া মাত্রই তথ্য পাওয়া যায়, সে সব স্মৃতিকে র‍্যানডম অ্যাক্শেস বলা হয়। একমাত্র ইলেক্ট্রনিক RAM, ROM এবং কোর মেমরিই হলো সম্পূর্ণ র‍্যানডম অ্যাক্শেস স্মৃতি ভাণ্ডার। অসম্ভব দ্রুত তথ্য লেখা এবং পড়া যায় বলে কম্পিউটারের আভ্যন্তরীন স্মৃতি ভাণ্ডার বা Primary Storage হিসেব র‍্যানডম অ্যাক্শেস স্মৃতি ব্যবস্থা ব্যবহার করা হয়। হার্ড ডিস্ক এবং ডিসকেট্ বা ফ্লপিকে অনেক সময় ভুল করে র‍্যানডম অ্যাক্শেস বলা হয়। কিন্তু প্রকৃতপক্ষে এগুলি সবই ডাইরেক্ট অ্যাক্শেস স্মৃতি ভাণ্ডার।

ডাইরেক্ট অ্যাক্শেস

ডাইরেক্ট অ্যাকশেস (Direct access) কিছুটা র‍্যানডম অ্যাক্শেসেরই মতো। যে সব স্মৃতিতে এই ব্যবস্থা থাকে সেই সব স্মৃতির যে কোনো মেমরি লোকেশনে তথ্য পড়া কিম্বা লেখা যায়। তবে তথ্য পড়তে চাওয়া মাত্রই তথ্য পাওয়া যায় না। এর জন্য সামান্য সময় অপেক্ষা করতে হয়। ফ্লপি, হার্ড ডিস্ক ইত্যাদি হলো এই ধরনের স্মৃতি ব্যবস্থা। দ্রুততার দিক থেকে র‍্যানডম অ্যাক্শেসের পরই ডাইরেক্ট অ্যাক্শেসের স্থান।

সিরিয়াল অ্যাক্শেস

ম্যাগনেটিক টেপ বা ডিজিটাল ক্যাসেট রেকর্ডার থেকে তথ্য পড়তে আরো দেরী হয়। এর কারণ, এই ধরনের স্মৃতিতে তথ্যগুলি একটানা (Serially) রেকর্ডিং করা হয়। ফলে কোনো বিশেষ তথ্য পড়ে ফেলার ব্যাপারে ডাইরেক্ট অ্যাক্শেস করা সম্ভব হয় না (ঠিক যেমন ক্যাসেটে রেকর্ডিং করা বহু গানের মধ্যে থেকে কোন একটি বিশেষ গান একবারে খুঁজে বার করা যায় না)।

পরিচ্ছেদ ৬

কম্পিউটার মস্তিষ্ক C. P. U

কম্পিউটারের প্রধান গণনা কেন্দ্র বা C. P. U কে আমাদের মস্তিষ্কের সঙ্গে তুলনা করা চলে। সেন্ট্রাল প্রসেসিং ইউনিট (Central Processing Unit) কথা থেকে C. P. U শব্দের উৎপত্তি। ইনপুট, আউটপুট, স্মৃতি ভাণ্ডার এবং কম্পিউটারের অন্যান্য সমস্ত সার্কিট C. P. U ই নিয়ন্ত্রণ করে। তাছাড়া কম্পিউটারের মাধ্যমে যে সব গণনা কার্য করা হয়, তা সবই C. P. U করে থাকে।

তবে C. P. U র একা কোনো কাজ করার ক্ষমতা নেই। ROM এ লিপিবদ্ধ Firmware (কম্পানি থেকে ভরে দেওয়া C. P. U নিয়ন্ত্রক প্রোগ্রাম) এবং কম্পিউটার ব্যবহারকারীর দেওয়া প্রোগ্রাম থেকে নির্দেশ গ্রহণ করে C. P. U তার যাবতীয় গণনা কার্য এবং ইনপুট, আউটপুট ডিভাইসকে নিয়ন্ত্রণ করার কাজ সম্পন্ন করে।

শুনতে কম্পিউটারের মস্তিষ্ক হলেও C. P. U আসলে অনেকগুলি বিভিন্ন ইলেক্ট্রনিক সার্কিটের সমষ্টি বিশেষ। একটি C. P. U র মধ্যে ছোট ছোট ফ্লিপ ফ্লপ RAM স্মৃতি থাকে, যাকে বলা হয় রেজিস্টার (Register)। এছাড়া থাকে ROM এবং RAM এর বিভিন্ন মেমরি লোকেশন অ্যাড্রেশ (Memory location address) করার জন্য ইলেক্ট্রনিক কাউন্টার (counter), বাইনারি যোগ এবং বিয়োগ করার ইলেক্ট্রনিক সার্কিট ALU (Arithmatic

৮

C. P. U

Logic Unit)। তথ্য আদান প্রদানের জন্য SIPO এবং PISO শিফট্ রেজিস্টার (Shift Register)। কম্পিউটারের বিভিন্ন অংশের কাজকর্ম সমান তালে চালাবার জন্য বৈদ্যুতিক স্পন্দক বা clock generator এবং ইনপুট, আউটপুট, RAM, ROM কে চালু বা বন্ধ করার নিয়ন্ত্রণ সার্কিট।

বর্তমানে V. L. S. I টেকনোলজির দৌলতে C. P. U সার্কিটকে একটি বড় লম্বা আইসির মধ্যে পুরে ফেলা সম্ভব হয়েছে। এগুলিকে বলা হয় মাইক্রোপ্রসেসর (এর থেকেই মাইক্রো কম্পিউটার কথাটির উৎপত্তি)।

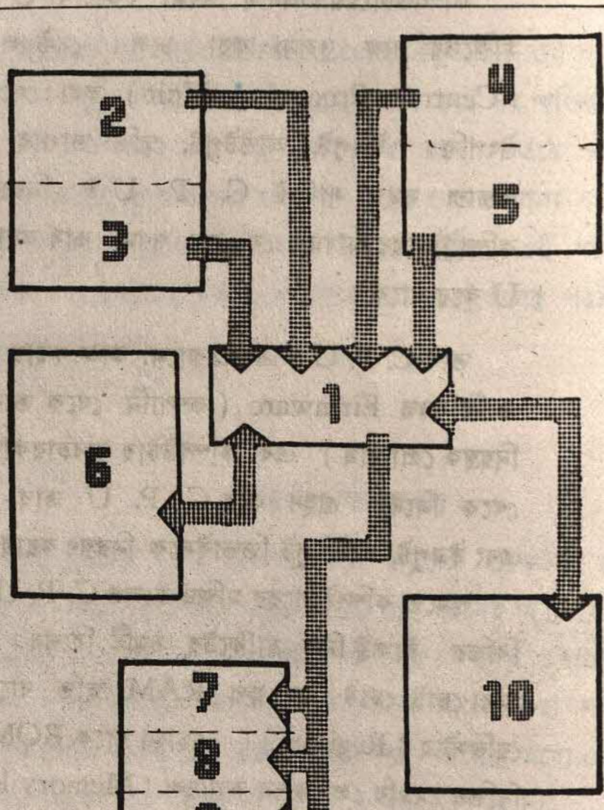
একটি আধুনিক মাইক্রোকম্পিউটারের ব্লক ডায়াগ্রাম আঁকলে সহজেই বোঝা যায় C. P. U কি ভাবে সমস্ত কম্পিউটারকে নিয়ন্ত্রণ করে এবং একই সঙ্গে কি ভাবে ROM এ রক্ষিত Firmware প্রোগ্রাম, C. P. U কে নিয়ন্ত্রণ করে সমস্ত কাজ করিয়ে নেয়।

চিত্র ৪১

একটি আধুনিক কম্পিউটারের ব্লক ডায়াগ্রাম।

ছবিতে দেখানো হয়েছে কিভাবে বিভিন্ন পেরিফেরালস্ থেকে C P.U এ তথ্য যাতায়াত করে।

1 C. P. U
2 কী বোর্ড
3 অন্যান্য ইনপুট ডিভাইস
4 ROM
5 সেকেন্ডারী স্মৃতি ভান্ডার
6 RAM
7 V. D. U
8 প্রিন্টার
9 স্পিচ্ সিন্থেসাইজার
10 মডেম



I/O এবং BUS

কম্পিউটার BUS ব্যবস্থা

C. P. U এর কার্যপ্রণালী বোঝবার আগে প্রথমেই কম্পিউটার বাস (BUS) সম্বন্ধে পরিষ্কার ধারণা থাকা দরকার, কারণ কম্পিউটারের যাবতীয় কাজকর্ম নিয়ন্ত্রণ, তথ্য আদান প্রদান ইত্যাদি সবই C. P. U ওই বাসের সাহায্যে করে থাকে।

কম্পিউটারের গায়ে লাগানো থাকে একটি সকেট যার নাম I/O Port (Input/Output Port). I/O Port হলো ইনপুট ডিভাইস থেকে তথ্য নিয়ে যাওয়া এবং কম্পিউটার থেকে আউটপুট ডিভাইসে তথ্য বয়ে আনার সকেট। সাধারণত বড় মাইক্রো কম্পিউটারে একাধিক I/O সকেট থাকে। এই I/O পোর্টের মাধ্যমেই কম্পিউটারের সঙ্গে প্রিন্টার বাড়তি স্মৃতি ভাণ্ডার, মনিটর প্রভৃতি পেরিফেরালস্ লাগানো হয়।

Data Bus

I/O সকেট থেকে যে তার গুচ্ছ বেরিয়ে আসে, তাকে বলা হয় ডেটা বাস (Data Bus)। Data . Bus মারফৎ ইনপুট থেকে কম্পিউটারে এবং কম্পিউটার থেকে আউটপুটে তথ্য আদান প্রদান হয়। মজার কথা হলো আধুনিক কম্পিউটারে একই তার গুচ্ছের মাধ্যমে ইনপুট ডিভাইস বা RAM/ROM স্মৃতি থেকে C. P. U তথ্য পড়ে আর আউটপুট ডিভাইস কিম্বা RAM স্মৃতিতে তথ্য প্রেরণ করে। সুতরাং Data Bus মারফৎ উভয় দিকেই তথ্য যাতায়াত করে। তাই Data Bus কে কম্পিউটার পরিভাষায় উভমুখী বা Bidirectional বলা হয়। এই ব্যবস্থার ফলে একই Data Bus এর সঙ্গে C. P. U, RAM, ROM এবং I/O port controller peripheral IC গুলি লাগানো হয়।

Control Bus, trisate logic

এখন প্রশ্ন হলো C. P. U কখন I/O Port, RAM বা ROM স্মৃতি থেকে তথ্য পড়বে কিম্বা তথ্য প্রেরণ করবে, তা অন্য সব সার্কিট-গুলি বুঝবে কেমন করে? এর জন্য C. P. U থেকে RAM, ROM এবং I/O Port controller peripheral ICগুলিতে আলাদা আলাদা নিয়ন্ত্রণ তার পাঠানো হয়। এই তার গুচ্ছের মাধ্যমে

C. P. U থেকে নিয়ন্ত্রণ নির্দেশ কম্পিউটারের বিভিন্ন অংশে ছড়িয়ে পড়ে। কম্পিউটার পরিভাষায় এই তার গুচ্ছকে বলা হয় কন্ট্রোল বাস (Controle Bus)।

Tristate logic

C. P. U কোনো সার্কিটের সঙ্গে যোগাযোগ স্থাপন করতে চাইলে, Control Bus এর মাধ্যমে সেই সার্কিটকে সচল করার নির্দেশ পাঠায়। ধরা যাক যদি RAM ভাণ্ডার থেকে তথ্য পড়ার দরকার হয়, তবে C. P. U থেকে RAM ভাণ্ডারের সঙ্গে যুক্ত Control Busটির মাধ্যমে RAM চালু করার নির্দেশ পাঠানো হয়। নির্দেশটি পৌঁছলে, তবেই RAM ভাণ্ডার থেকে তথ্য পড়া বা লেখা যায়। কিন্তু কম্পিউটারের যাবতীয় RAM, ROM I/O সার্কিট সবই একত্রে একই Data Bus এর সঙ্গে সংযুক্ত থাকায় কোনো একটি RAM এ তথ্য লেখা বা পড়া হলে অন্যান্য সার্কিটেও সেই একই তথ্য গিয়ে পৌঁছয়। এটা আমাদের কাম্য নয়। তাই কম্পিউটারের RAM, ROM, I/O প্রভৃতি গুলিকে এমন ভাবে তৈরী করা হয় যে তাদের Controle Bus এ চালু হবার নির্দেশ না পাঠালে তারা সর্বদা Data Bus থেকে বৈদ্যুতিকভাবে বিচ্ছিন্ন হয়ে থাকে।

অর্থাৎ সাধারণ অবস্থায় সমস্ত ICর পিন গুলি Data Bus এর থেকে বৈদ্যুতিক সংযোগ হারিয়ে ভাসমান (Floating) অবস্থায় থাকে। ডিজিটাল সংকেতে কিন্তু এই ভাসমান বা Floating state এর কোনো সংজ্ঞা নেই।

ডিজিটাল সংকেত অনুসারে কোনো পরিবাহী 0 ভোল্টে থাকলে তা বাইনারি 0 বোঝায় আর 2 ভোল্টের বেশী বিভবে থাকলে তা বাইনারি 1 বোঝায়। কিন্তু বিদ্যুৎ সংযোগ বিহীন অবস্থা কোনো বাইনারি সংখ্যাই বোঝায় না। কম্পিউটার পরিভাষায় এই Floating অবস্থাকে Tristate বা তৃতীয় অবস্থা বলে অভিহিত করা হয়।

Data Bus এর সঙ্গে সংযুক্ত সমস্ত ইলেক্ট্রনিক সার্কিটই Tristate device করতে হয়, ফলে কম্পিউটারে ব্যবহৃত যাবতীয় RAM, ROM, I/O Port Controller এবং C. P. U সবেতেই Tristate এর ব্যবস্থা থাকে।

ADDRESS BUS

C. P. U থেকে আর এক গুচ্ছ তার বেরিয়ে এসে আভ্যন্তরীণ স্মৃতি ভাণ্ডারের সঙ্গে যুক্ত হয়, এরা হলো অ্যাড্রেশ বাস (Address Bus)। এর সাহায্যে C. P. U, ROM এবং RAM স্মৃতির বিভিন্ন Memory location এর সঙ্গে সংযোগ স্থাপন করে সেই জায়গায় রক্ষিত তথ্য Data Bus এর মাধ্যমে পড়ে নেয়, কিম্বা নোতুন তথ্য লিখে রাখে।

Data Bus এর মতো Address Bus কিম্বা Control Bus কিন্তু Bidirectional নয়। Address Bus এবং Control Bus এর মাধ্যমে সর্বদা C. P. U থেকে তথ্য অন্যান্য অংশে ছড়িয়ে পড়ে।



অ্যাড্রেশ বাস

যে কোন সমস্যা সমাধান করতে C.P.U-কে সর্বদা কম্পিউটারের আভ্যন্তরীণ স্মৃতি ভাণ্ডারের সাহায্য নিতে হয়। C.P.U এবং আভ্যন্তরীণ স্মৃতি ভাণ্ডারের যৌথ প্রচেষ্টায় কম্পিউটার সমস্ত গণনা কার্য করে থাকে। একটি ছোট উপমা দিয়ে কম্পিউটারের কার্যপ্রণালী বোঝানো যেতে পারে। C.P.U-র সঙ্গে যদি ট্যাক্সির তুলনা করা যায় তবে ট্যাক্সি ড্রাইভার হলো আভ্যন্তরীণ স্মৃতি ভাণ্ডার এবং সওয়ারী হলো প্রোগ্রাম। সওয়ারীর নির্দেশ মতো যেমন ড্রাইভার ট্যাক্সি চালিয়ে সওয়ারীকে সঠিক গন্তব্যস্থলে পেঁছে দেয় তেমনি প্রোগ্রামের নির্দেশ অনুযায়ী ROM স্মৃতিতে লেখা ইন্টার প্রিন্টার, কম্পাইলার এবং ফার্ম ওয়্যার ট্যাক্সি ড্রাইভারের মতো C.P.U-কে চালিয়ে, অর্থাৎ C.P.U-কে দিয়ে সঠিক গণনা করিয়ে নিয়ে ফলাফল জানিয়ে দেয়।

C. P. U কি করে কাজ করে

ROM এর Firmware ছাড়া C. P. U যে সম্পূর্ণ অচল একথা আমরা পূর্বেই জেনেছি। এবার দেখা যাক কি ভাবে C. P. U, ROM এর Firmware এর সাহায্যে কোনো একটি কাজ সম্পন্ন করে।

সেই আগের উদাহরণটিই ধরা যাক। কম্পিউটার অন্ করে কী প্যাডের A বোতামটি টেপা হলো। অমনি সঙ্গে সঙ্গে V. D. U স্ক্রিনে কম্পিউটার A হরফটি ফুটিয়ে তুললো। এই ছোট কাজটি কি ভাবে C P. U করে থাকে, সেটিই বিশদ ভাবে বিশ্লেষণ করে দেখা যাক।

(১) কম্পিউটার অন্ করার সঙ্গে সঙ্গে কম্পিউটারের ভিতরে তৈরী করা স্বয়ংক্রিয় Reset সার্কিট C. P. U এর আভ্যন্তরীণ Register (কোনো কোনো সময় accumulator ও বলা হয়) এবং RAM স্মৃতি ভাণ্ডারের প্রতিটি মেমরি সেলে বাইনারি 0 তথ্য লিখে দেবে এবং C. P. U র Memory location address Counter কে 0000 (Hex) এ সেট করে দেবে।

(২) C. P. U র Memory location Counter এরপর ROM স্মৃতিভাণ্ডারের 0001 (Hex) মেমরি লোকেশন অ্যাড্রেশ করবে এবং একই সঙ্গে কন্ট্রোল বাসের মাধ্যমে ROM স্মৃতিকে চালু করবে।

৮

C. P. U এর কার্যপ্রণালী

(৩) ROM স্মৃতির Firmware এর 0001 মেমরি লোকেশনে 8 বিট্ বাইনারী সংখ্যার মাধ্যমে মেশিন ল্যাংগোয়েজে 'তথ্য পড়ো' (Load) নির্দেশ লেখা আছে। এই তথ্য সঙ্গে সঙ্গে Data Bus এ চলে আসবে।

(৪) C. P. U 'তথ্য পড়ো' নির্দেশটি Date Bus থেকে গ্রহণ করে তার অভ্যন্তরের একটি Register বা accumulator এ লিখে রাখবে। এই Register বা accumulator কে কম্পিউটার পরিভাষায় বলা হয় Instruction Register C. P. U মেশিন ল্যাগোয়েজে লেখা নির্দেশটি Decode করে বুঝে নেবে তাকে কি নির্দেশ দেওয়া হয়েছে।

(৫) ROM এর Firmware থেকে C. P. U কে শুধু 'তথ্য পড়ো' দেওয়া হয়েছে। অথচ কোথা থেকে তথ্য পড়তে হবে তা বলা হয়নি। তাই C. P. U এর Memory location address Counter স্বয়ংক্রিয় ভাবে আর এক ধাপ বেড়ে ROM স্মৃতির 0002 (Hex) মেমরি লোকেশন অ্যাড্রেশ করবে এবং কন্ট্রোল বাসের সাহায্যে ROM কে Tristate থেকে চালু করবে।

(৬) কোম্পানী থেকে ROM স্মৃতির 0002 লোকেশনে লেখা আছে কোনো একটি বিশেষ I/O Port এর সংকেতিক বাইনারি নাম্বার। ধরা যাক এটি হলো I/O Port 1। সুতরাং 0002 অ্যাড্রেশ করা মাত্র 1 সংখ্যাটির বাইনারি তথ্য পাওয়া যাবে।

(৭) C. P. U ডেটা বাস থেকে তথ্য পড়ে সেটি পূর্বের Instruction Register এ লিখে ফেলবে। এবার তাহলে সম্পূর্ণ নির্দেশটি হবে—'I/O Port 1 থেকে তথ্য পড়ো'।

(৮) এই সম্পূর্ণ নির্দেশটি Decoding করে C. P. U 1 নাম্বার I/O Port কে কন্ট্রোল বাসের মাধ্যমে Tristate থেকে চালু করবে।

ধরা যাক আমাদের কী বোর্ডটি I/O Port 1 এর সকেটেই গোঁজা আছে। যতক্ষণ না কী বোর্ডের কোনো বোতাম টেপা হবে, ততক্ষণ I/O Port 1 থেকে কোনো তথ্য আসবেনা। তাই C. P. U চুপচাপ অপেক্ষা করে বসে থাকবে কতক্ষণে আমরা কী বোর্ডের একটি বোতাম টিপবো।

এবার মনে করুন আমরা কী বোর্ডের A লেখা বোতামটি টিপলাম।

(৯) সঙ্গে সঙ্গে I/O Port 1 এর সঙ্গে লাগানো ডেটা বাসে A হরফের বাইনারি কোড চলে আসবে।

(১০) C. P. U কোডটি ডেটা বাস থেকে পড়ে তার অভ্যন্তরের ছোট্ট RAM স্মৃতিতে (accumulator) লিখে রাখবে।

(১১) তথ্য লেখা শেষ হলেই C. P. U, ROM স্মৃতির 0003 (Hex) মেমরি লোকেশন অ্যাড্রেশ করে ROM কে চালু করবে। এই লোকেশনে Firmware হিসেবে 'তথ্য সংরক্ষণ করো' (Store) নির্দেশটি মেশিন ল্যাংগোয়েজে লেখা আছে।

(১২) তথ্যটি সঙ্গে সঙ্গে ডেটা বাসে বেরিয়ে আসবে।

(১৩) C. P. U নির্দেশটি আবার পূর্বের মতো Instruction Register এ লিখে নেবে।

(১৪) এবারও কিন্তু C. P. U কে সম্পূর্ণ নির্দেশ দেওয়া হয়নি। অর্থাৎ কোথায় তথ্যটি লিখতে হবে তা বলা হয়নি। তাই C. P. U ROM স্মৃতির 0004 (Hex) মেমরি লোকেশন আড্রেশ করবে আর কন্ট্রোল বাস মারফৎ ROM কে পুনরায় চালু করবে।

(১৫) ROM চালু হলেই 0004 লোকেশনে লেখা RAM ভাণ্ডারের প্রথম মেমরি লোকেশনটির তথ্য ডেটা বাসে পাওয়া যাবে। ধরা যাক এটি হলো 8000 (Hex)।

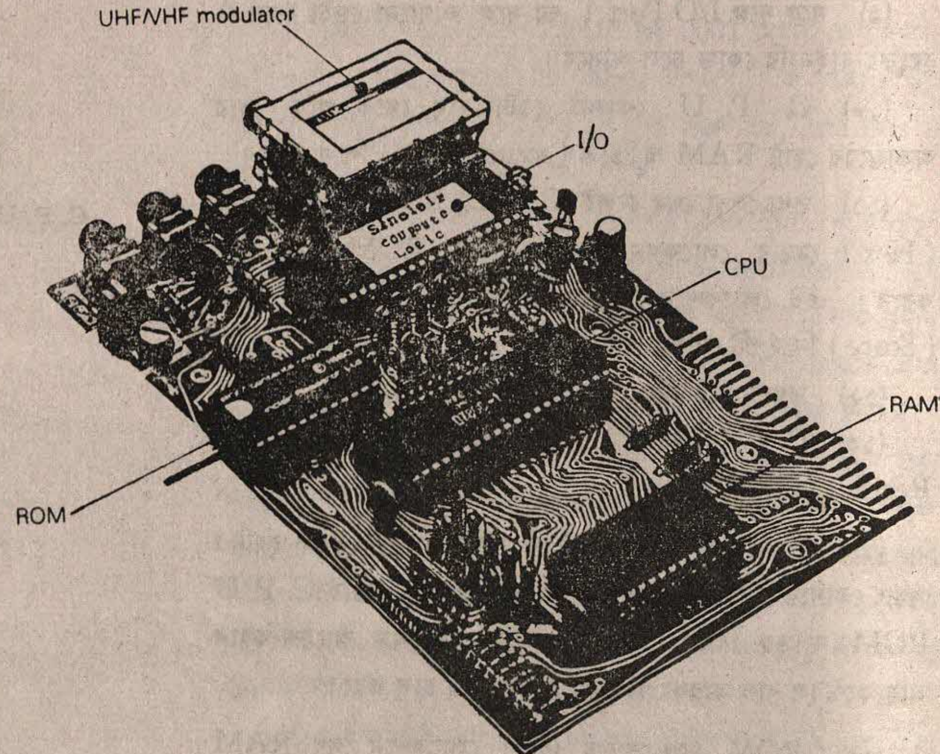
(১৬) C. P. U পূর্বের মতো Instruction Register এ 8000 (Hex) তথ্যটি লিখবে। ফলে পুরো নির্দেশটি দাঁড়াবে '8000 (Hex) মেমরি লোকেশনে accumulator এর তথ্য সংরক্ষণ করো'।

(১৭) তথ্যটি Decode করে C. P. U কন্ট্রোল বাস মারফৎ RAM চালু করবে এবং 8000 (Hex) মেমরি লোকেশন অ্যাড্রেশ করবে। একই সঙ্গে অপর একটি কন্ট্রোল বাসের সাহায্যে RAM স্মৃতিকে 'তথ্য লেখো' (Write Mode) অবস্থায় আনবে।

(১৮) এরপর ডেটা বাসের সঙ্গে C. P. U তার আভ্যন্তরীণ accumulator এর সংযোগ ঘটিয়ে দেবে। ফলে ডেটা বাসে A হরফটির ডিজিটাল তথ্য চলে আসবে এবং RAM স্মৃতিতে সেটি Copy হয়ে যাবে। এই প্রক্রিয়ায় কিন্তু accumulator এর তথ্যটি আবিকৃতই থেকে যাবে।

৮

C. P. U এর কার্যপ্রণালী



চিত্র ৪২

একটি ছোট পারশোনাল কম্পিউটার খুলে দেখানো হয়েছে কি ভাবে তার অভ্যন্তরে বিভিন্ন ইলেকট্রনিক যন্ত্রাংশ লাগানো আছে।

(১৯) পরবর্তী নির্দেশ জানতে C. P. U পুনরায় ROM স্মৃতির সঙ্গে সংযোগ স্থাপন করবে। এবার অবশ্য 0005 (Hex) মেমরি লোকেশন অ্যাড্রেশ করা হবে।

(২০) কোম্পানী থেকে ওই স্থানে "accumulator এর তথ্য পাঠাও" নির্দেশ ভরা আছে। এই তথ্য ROM থেকে ডাটা বাস মাধ্যমে C. P. U তে এসে পৌঁছবে।

(২১) তথ্যটি C. P. U তার Instruction Register এ লিখে নেবে।

(২২) C. P. U পুনরায় কোথায় তথ্য পাঠাতে হবে তা জানার জন্য ROM এর 0006 (Hex) মেমরি লোকেশন অ্যাড্রেশ করবে।

(২৩) এই লেকেশন C. R. T Controller Peripherals এর সংকেত লেখা আছে।

(২৪) ডেটা বাস থেকে এই সংকেত C. P. U পূর্বের Instruction Register এ লিখে ফেলবে। সুতরাং পুরো নির্দেশটি হবে "অ্যাকুমুলেটরের তথ্য C. R. T Controller I. C তে পাঠাও"।

(১৫) নির্দেশটি Decode করে C. P. U, C. R. T Controller I. C কে কন্ট্রোল বাস মারফৎ চালু করে A হরফের তথ্য পাঠিয়ে দেবে।

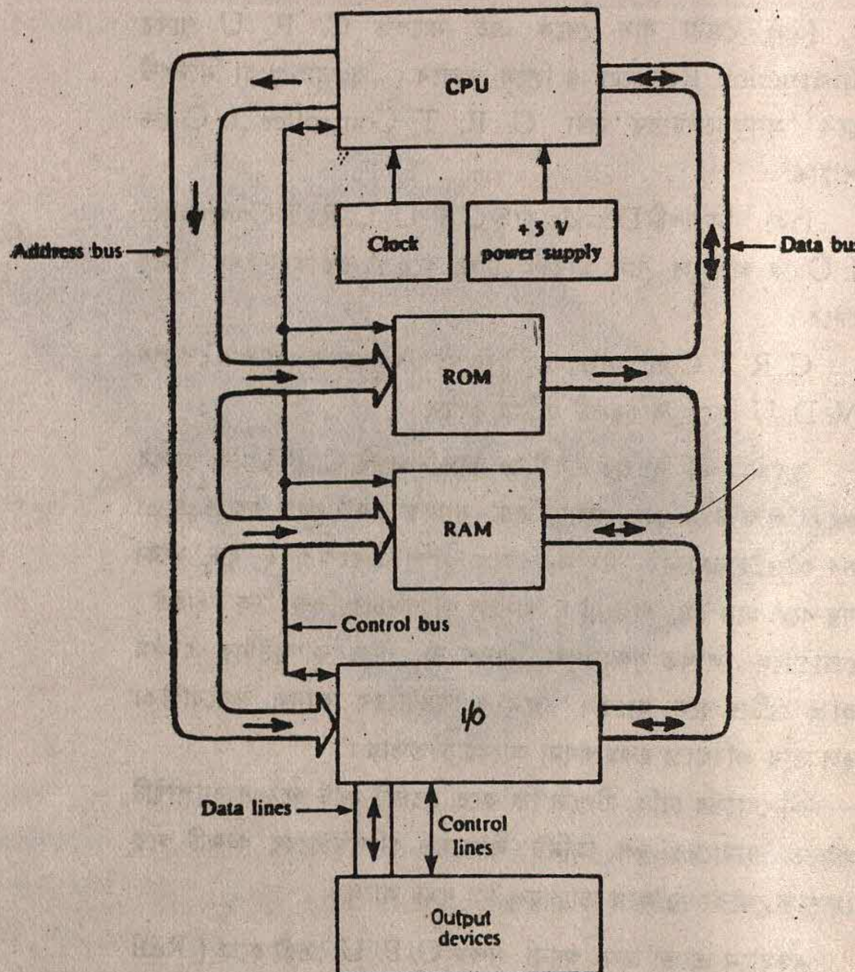
C. R T Controller I. C টি ওই A হরফের সংকেত পেলেই V. D. U স্ক্রিনে A হরফটি ফুটিয়ে তুলবে।

শুধু মাত্র কী বোর্ডের একটি বোতাম টিপলেই C. P. U কে পরপর এতো গুলি কাজ করে ফেলতে হয়। সামান্য একটি কাজ করতে এতো সব কাণ্ড কারখানার প্রয়োজন দেখে কম্পিউটারকে খুবই 'বুদ্ধু' মার্কা যন্ত্র বলে মনে হয়, তাই না? আসলে কম্পিউটার কিন্তু ঠিক তাইই! প্রোগ্রামে লিখিত পুঙ্খানুপুঙ্খ নির্দেশ না পেলে কম্পিউটার একদম হাত গুটিয়ে বসে থাকে। তবে কম্পিউটারের আসল উপযোগীতা হল তার অবিশ্বাস্য রকম গণনা কার্যের দ্রুততায়।

কী বোর্ডের চাবি টিপলে কি করে হরফটি ফুটে ওঠে এ ব্যাপারটি পড়তে আমাদের দশ মিনিট লাগলেও কম্পিউটারের কাজটি করে ফেলতে সেকেণ্ডের লাখ ভাগেরও কম সময় লাগে।

বর্তমানে এতো দ্রুত গণনা সক্ষম C. P. U তৈরী হচ্ছে (Real time Operation) যাতে তথ্য পাঠাবার সঙ্গে সঙ্গেই উত্তর পাওয়া যায়। এই সমস্ত কম্পিউটার তাই কোন এক ব্যাক্তির ব্যবহারের পক্ষে উপযোগী নয় কারণ আমরা যে গতিতে তথ্য প্রোগ্রাম করতে পারি বা পড়তে পারি, তার থেকে ঢের দ্রুত কম্পিউটারটি গণনা করতে পারে। সেই জন্য এই সব যুক্ত কম্পিউটারকে বিশেষ ইলেক্ট্রনিক ব্যবস্থার সাহায্যে পর্যায়ক্রমে বহু কম্পিউটার ব্যবহারকারীর সঙ্গে প্রতি সেকেণ্ডে একশো ভাগেরও কম সময়ের জন্য যুক্ত করা হয়। ওই অল্প সময়ের মধ্যেই C. P. U যে ব্যবহারকারীর সঙ্গে যুক্ত হয়, তার যাবতীয় গণনা সমাধান করে দেয়। ফলে সব কজন কম্পিউটার ব্যবহারকারীই এক সঙ্গে একই C. P. U র মাধ্যমে আলাদা আলাদা সমস্যা সমাধান করে নিতে পারেন। এই ব্যবস্থাকে বলা

C. P. U এর কার্যপ্রণালী



চিত্র ৪০

কম্পিউটার অভ্যন্তরস্থ BUS ব্যবস্থার ব্লক ডায়াগ্রাম। এই ডায়াগ্রামে দেখানো হয়েছে কিভাবে C. P. U. বিভিন্ন BUS এর মাধ্যমে সমস্ত Peripheralsদের নিয়ন্ত্রণ করে।

হয় Time Sharing ব্যবস্থা। বিদেশে এই ব্যবস্থা খুবই জনপ্রিয়। টেলিফোন বা ইলেট্রিক সাপ্লাইয়ের গ্রাহকের মতোই Time Sharing terminal এর গ্রাহক হওয়া যায়। মূল C. P. U এবং কম্পিউটারটি শহরের প্রধান কম্পিউটার ভবনে বসানো থাকে। কম্পিউটারের সঙ্গে গ্রাহকের সাধারণ টেলিফোন লাইনের মাধ্যমে যোগাযোগ থাকে। গ্রাহকের কাছে থাকে একটি কী বোর্ড, V. D. U এবং Modem। কী বোর্ড এবং V. D. U কে গ্রাহক Modem এর মাধ্যমে বাড়ির টেলিফোনের সঙ্গে যুক্ত করে টেলিফোনে কম্পিউটারের নম্বর ডায়াল করেন। কম্পিউটারের সঙ্গে সংযোগ স্থাপন হলেই কম্পিউটারের C. P. U মডেমের সাহায্যে ডিজিটাল তথ্যকে Serial data হিসেবে গ্রাহকের বাড়ি পাঠিয়ে দেয়। এর ফলে গ্রাহকের রিসিভারে শব্দ স্পন্দন সৃষ্টি হয়। গ্রাহকের বাড়ির Modem সেই শব্দ থেকে পুনরায় ডিজিটাল সংকেত পুনরুদ্ধার করে V. D. U তে লেখা ফুটিয়ে তোলে। গ্রাহক এবার কী বোর্ডে তাঁর গ্রহক নাম্বার টাইপ করে C. P. U র সঙ্গে সংযোগ স্থাপন (যাকে কম্পিউটার পরিভাষায় বলে Log In) করেন। এরপর সাধারণ কম্পিউটার যে ভাবে ব্যবহার করতে হয়, সেই ভাবে কী বোর্ড এবং V. D. U র সাহায্যে গ্রাহক তাঁর কাজ কর্ম সেরে নেন। কাজ শেষ হলে, কতক্ষণ এবং কী ধরনের জটিল গণনা করা হলো, সেই অনুসারে স্বয়ংক্রিয় ভাবে কম্পিউটারের C. P. U বিল তৈরী করে দেয়।



C. P U এর কার্যপ্রণালী

এই ব্যবস্থা আমাদের দেশেও অদূর ভবিষ্যতে চালু হবে। ইতিমধ্যেই বড় বড় ভারতীয় বাণিজ্যিক প্রতিষ্ঠান তাঁদের নিজস্ব কাজকর্মের জন্য এই ধরনের কম্পিউটার বসিয়েছেন।

C.P.U সংক্রান্ত কয়েকটি দরকারি

বিভিন্ন ধরনের BUS এবং C. P. U এর কার্য প্রণালি সম্বন্ধে আমাদের জানা হয়ে গেছে। কিন্তু কিভাবে এই BUS এর মাধ্যমে C. P. U থেকে I/Oতে তথ্য যাতায়াত করে সেটা না জানলে এ বিষয়ে জ্ঞান অসম্পূর্ণ থেকে যাবে। খুব সংক্ষেপে সেই বিষয়গুলিই আমরা এবার জেনে নেব।

৮

সিরিয়াল/প্যারালাল ইণ্টারফেস

সিরিয়াল ইণ্টারফেস

দুটি তারের মাধ্যমে কম্পিউটারের সঙ্গে অন্যান্য পেরিফেরালসের (Peripherals) তথ্য আদান প্রদানকারী সংযোগ ব্যবস্থা হলো সিরিয়াল ইণ্টারফেস (Serial interface)। এই ব্যবস্থায় সমস্ত বাইনারি সংখ্যাকে এক বিট্ (Bit) করে ভেঙে ধারাবাহিকভাবে পাঠানো হয়। যেমন একটি ৪ বিট্ বাইনারি সংখ্যা ভাবা যাক ধরুন সেটি হলো 11010001 ; এই সংখ্যাটি serial ভাবে পাঠাতে গেলে 1, 1, 0, 1, 0, 0, 0, 1, হিসেবে এক এক করে পাঠাতে হবে। ম্যাগনেটিক টেপ কিম্বা ফ্লপির কথা চিন্তা করুন। এতে তথ্য লেখা বা পড়ার ব্যাপারে serial ব্যবস্থা গ্রহণ করা ছাড়া আর কোনো উপায় নেই। কারণ চৌম্বক পদার্থের কেবল 1 বা 0 ছাড়া আর কোনো সংখ্যা লেখার ব্যবস্থা নেই। সুতরাং ডিস্কে বা ফিতেতে 11010001 লিখতে হলে পরপর 1, 1, 0, 1, 0, 0, 0, 1 তথ্য রেকর্ডিং করতে হবে। তারপর ওই তথ্য পড়ার সময় 1, 1, 0, 1, 0, 0, 0, 1 হিসেবে ধারাবাহিকভাবে তথ্য পড়ে নিতে হবে।

V. D. U, প্রিন্টার, মডেম প্রভৃতি আউটপুট পেরিফেরালস্ গুলি কম্পিউটারের I/O পোর্টের সঙ্গে সিরিয়াল ইণ্টারফেস ব্যবস্থার মাধ্যমেই যুক্ত থাকে।

প্যারালাল ইণ্টারফেস

RAM বা ROM স্মৃতিতে প্যারালাল ইণ্টারফেসের (Parallel interface) ব্যবস্থা থাকে। এই ব্যবস্থায়, C. P. U এক সঙ্গে যতো বিট্ বাইনারি সংখ্যা পড়তে পারে, স্মৃতি থেকে ততোগুলি বৈদ্যুতিক তার data bus হিসেবে বেরিয়ে আসে। যেমন 8 bit মাইক্রো কম্পিউটারের C. P. U এক সঙ্গে 8 bit বাইনারি সংখ্যা পড়ে। এই ধরনের মাইক্রো কম্পিউটারের আভ্যন্তরীণ স্মৃতি থেকে তাই 8টি data bus C. P. U সঙ্গে সংযুক্ত হয়। C. P. U যখনই কোনো একটি মেমরি লোকেশন অ্যাড্রেশ করে, অমনি সেই মেমরি লোকেশন লিখিত 8 bit সংখ্যাটি একসঙ্গে 8টি তারের মাধ্যমে C. P. Uতে এসে পৌঁছয়।

সুতরাং বুঝতেই পারছেন প্যারালাল তথ্য আদান প্রদান ব্যবস্থার (Parallel data transfer) থেকে দ্রুততম ব্যবস্থা আর সম্ভব নয়।

পরিচ্ছেদ ৭

আউটপুট ডিভাইস

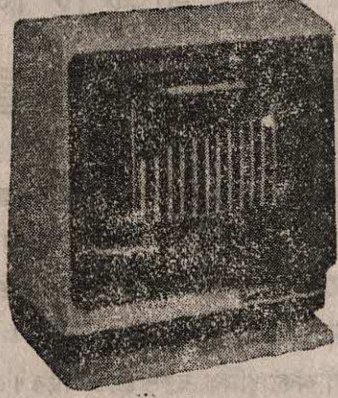
আউটপুট ডিভাইসের মাধ্যমে কম্পিউটার তার ফলাফল জানিয়ে দেয়। এটি সাধারণ ক্যালকুলেটরের এল. ই. ডি কিম্বা এল. সি. ডি ডিসপ্লে থেকে শুরু করে ভিডিও ডিসপ্লে ইউনিট, ম্যাট্রিক্স প্রিণ্টার, লাইন প্রিণ্টার, ইলেক্ট্রনিক টাইপ রাইটার, ইঙ্ক জেট্ প্রিণ্টারের মতো অত্যাধুনিক যন্ত্রও হতে পারে।

ভিডিও ডিসপ্লে ইউনিট

ভিডিও ডিসপ্লে ইউনিটের সংক্ষিপ্ত নাম ভি. ডি. ইউ (VDU)। সাধারণ টি ভিতে যা যন্ত্রাংশ লাগে, তার মধ্যে টিউনার, আই এফ, ডিসক্রিমিনেটর এবং অডিও সার্কিট বাদ দিলে যা পড়ে থাকে সেটিই হলো একটি ভি. ডি. ইউ (V.D.U)। এক কথায় V.D.U একটি অর্ধেক টিভি সেট যাতে দূরদর্শনের প্রোগ্রাম দেখা যায় না, কোনো শব্দও শোনা যায় না, কিন্তু একে যদি কম্পিউটার কিম্বা ভি. সি. আর-এর সঙ্গে লাগানো যায়, তবে এতে ছবি দেখা যায়। কালো সবুজ, কিম্বা রঙিন দু ধরণের মনিটর পাওয়া যায়। যে সব কম্পিউটারে ভিডিও ক্যারেক্টার জেনারেটর লাগানো আছে, তার সঙ্গে মনিটর লাগালে, কী বোর্ডে যে প্রোগ্রাম টাইপ করা হচ্ছে এবং কম্পিউটার যে ফলাফল জানাচ্ছে, তা সবই মনিটরের স্ক্রিনে ফুটে ওঠে।

ভি ডি ইউ/মনিটর

সাধারণত মাঝারি আকারের মাইক্রো কম্পিউটার এবং তার থেকে বড় সমস্ত কম্পিউটারের মনিটর লাগানোর ব্যাবস্থা থাকে। ছোটো পকেট কম্পিউটারে মনিটর লাগানো যায় না, কারণ ভি. ডি. ইউ স্ক্রিনে হরফ বা রেখা ফুটিয়ে তোলার জন্য প্রতি মুহূর্তে প্রচুর পরিমাণে তথ্য কম্পিউটার থেকে মনিটরে পাঠাতে হয়। এর জন্য একটি রীতিমতো বড় কম্পিউটার স্মৃতি ভাণ্ডারের দরকার পড়ে। বর্তমানে বেশীর ভাগ কম্পিউটারেই ভি. ডি. ইউকে পরিচালনার জন্য আলাদা ইলেকট্রনিক স্মৃতি ভাণ্ডারের ব্যবস্থা রাখা হয় যাতে কম্পিউটারের মূল স্মৃতি ভাণ্ডারকে কেবলমাত্র প্রোগ্রামিং কাজেই লাগানো যায়।



চিত্র ৪৪

VDU/মনিটর

হাই এবং লো রেজোলিউশন গ্রাফিক্স

শুধু মাত্র অ্যালফানিউমারিক ক্যারেক্টারই (হরফ, সংখ্যা এবং যোগ, বিয়োগ, গুণ, ভাগ ইত্যাদি চিহ্ন) নয়, আমরা ইচ্ছে করলে ভি. ডি. ইউ স্ক্রিনে সরল বা বক্ররেখাও আঁকতে পারি, যাকে কম্পিউটার পরিভাষায় বলে গ্রাফিক্স। গ্রাফিক্সের জন্য দরকার পড়ে অনেক বড় স্মৃতি ভাণ্ডার। গ্রাফিক্স দু ধরণের হয়, লো রেজোলিউশন (Resolution) এবং হাই রেজোলিউশন। হাই রেজোলিউশন গ্রাফিক্স-এর কাজ করার সময় কম্পিউটার সম্পূর্ণ টিভি স্ক্রিনকে অসংখ্য বিন্দুতে ভাগ করে নেয় (যেমন গ্রাফ পেপারের কাগজকে অনেক বর্গক্ষেত্রাকার অংশে ভাগ করে নেওয়া হয়)। লো রেজোলিউশন গ্রাফিক্সে এই বিন্দুর সংখ্যা হয় কম, অর্থাৎ স্ক্রিনে তাদের আকার হয় আরো বড়। ফলে

লো রেজোলিউশন গ্রাফিক্সের সাহয্যে সূক্ষ্ম লাইন, নিখুঁত ডিজাইন বা ছবি আঁকা সম্ভব নয়। হাই রেজালিউশন গ্রাফিক্সের সাহায্যে ভি. ডি ইউ স্ক্রিনে যে কোন সূক্ষ্ম ডিজাইন আঁকা সম্ভব। কম্পিউটার পরিভাষায় ভি. ডি. ইউ স্ক্রিনকে যে ছোট ছোট বিন্দুতে ভাগ করা হয়, তাকে বলে পিক্সেল (Pixel)। কোন একটি পিক্সেলকে আলোকিত করতে হলে কম্পিউটার স্মৃতিতে ওই পিক্সেলের কো-অর্ডিনেট জানিয়ে দিতে হয়। সবুজ কালো মনিটর লাগালে প্রতি পিক্সেলের জন্য দুটি বাইনারি তথ্য স্মৃতিতে জমা রাখতে হয়। আর যদি রঙিন গ্রাফিক্স তৈরী করতে হয়; তবে প্রতি পিক্সেলের জন্য চার বিট্ তথ্য জমা রাখতে হয়। সাধারণত হাই রেজোলিউশন গ্রাফিক্সের ক্ষেত্রে ভি· ডি. ইউ স্ক্রিনকে দশ থেকে পঞ্চাশ হাজার পিক্সেলে ভাগ করা হয়। এর থেকেই বোঝা যায় একটি হাই রেজোলিউশন কালার গ্রাফিক্স সৃষ্টি করতে গেলে কি বিরাট স্মৃতি ভাণ্ডার দরকার। এ সমস্ত সার্কিটই সাধারণত কালার গ্রাফিক্স কার্ড নামে একটি আলাদা বোর্ডে তৈরী করা থাকে। অনেক কম্পিউটারে এ ধরণের গ্রাফিক্স কার্ড বাড়তি সংযুক্তি হিসেবে লাগিয়ে নেওয়া যায়।

প্রিন্টার

মনিটর খুবই জনপ্রিয় হওয়া সত্ত্বেও এর থেকে কোনো ছাপানো উত্তর বা হার্ড কপি (Hard Copy) পাওয়া যায় না। যে সব ক্ষেত্রে শুধু উত্তরই জরুরি নয়, তার রেকর্ড রাখাও গুরুত্বপূর্ণ, সে সব জায়গায় মনিটরের এর সঙ্গে সঙ্গে প্রিন্টার (Printer)ব্যবহার করা হয়। প্রিন্টার হলো এমন একটি যন্ত্র যেটি কম্পিউটারের সঙ্গে লাগালে মনিটরে যে যে অ্যালফানিউমারিক ক্যারেক্টার ফুটে উঠছে তাকে তৎক্ষণাৎ কাগজে ছাপিয়ে ফেলে। কিছু কিছু এমন প্রিন্টারও আছে যাতে শুধু অ্যালফানিউমারিক ক্যারেকটারই নয়, যে কোন জটিল গ্রাফিক্সও ছাপানো যায়। বর্তমানে বহু কোম্পানি বহু ধরণের প্রিন্টার তৈরী করছেন। এর মধ্যে এক এক ধরণের প্রিন্টারের এক এক রকম বৈশিষ্ট্য আছে। কোনোটি অসম্ভব দ্রুত লেখা ছেপে ফেলতে পারে, কোনটি আকারে ছোট আর দামে সস্তা, কোনোটিতে প্রিনটিং প্রেসের মতো নিখুত ছাপা বা Letter Quality প্রিনটিং হয়,

লাইন প্রিন্টার

কেউ বা গ্রাফিক্স আঁকতে পারে ইত্যাদি। কার্যগত দিক থেকে সমস্ত প্রিনটারকে মোট পাঁচ ভাগে ভাগ করা যায়—

১। ইলেক্‌ট্রোমেকানিকাল প্রিন্টার
২। ইলেক্ট্রোম্যাগনেটিক প্রিন্টার
৩। ইলেক্ট্রোস্ট্যাটিক প্রিন্টার
৪। থার্মাল প্রিন্টার
৫। ইংক জেট প্রিন্টার

মাইক্রো কম্পিউটারে সব থেকে বেশী ব্যবহার করা হয় ইলেক্ট্রোমেকানিকাল ইম্প্যাক্ট কিম্বা থার্মাল প্রিন্টার, কারণ এ ধরণের দামে প্রিন্টার বেশ সস্তা। বড় কম্পিউটারে ব্যবহার হয় বিশেষ ধরণের ইলেক্ট্রোমেকানিকাল, ইলেক্ট্রোস্ট্যাটিক এবং ইংকজেট প্রিন্টার। এই তিন ধরণের প্রিন্টারের সব থেকে বড় সুবিধা হলো এরা অসম্ভব দ্রুত ছাপতে পারে।

লাইন প্রিন্টার

ইলেক্‌ট্রোমেকানিকাল প্রিন্টারের মধ্যে এগুলিই হলো সব থেকে দ্রুত এবং বড় কম্পিউটারে সর্বাধিক ব্যবহৃত প্রিন্টার। লাইন প্রিন্টার আবার অনেক ধরণের হয়। তবে সব থেকে বেশী প্রচলিত ধরণটির কথাই এখানে বলা হলো। এ ধরণের লাইন প্রিন্টারকে ব্যারেল প্রিন্টারও (Barrel Printer) বলা হয়।

ব্যারেল প্রিন্টারের প্রধান বৈশিষ্ট্য হলো এতে ছাপার হরফগুলি একটি ধাতব প্রিন্ট ব্যারেলের গায়ে বসানো থাকে। সিলিণ্ডারের মতো দেখতে ধাতব ব্যারেলটি লম্বায় যে কাগজে ছাপা হবে তার প্রস্থের সমান হয়। ব্যারেলের গায়ে অ্যালফানিউমারিক ক্যারেকটার-গুলি উঁচু উঁচু করে এক একটি লাইন বরাবর লাগানো। থাকে প্রতি লাইনে কেবল একই ধরণের ক্যারেকটার থাকে (যেমন ছবিতে দেখানো হয়েছে)। সাধারণত এক একটি লাইনে 120টি ক্যারেকটার থাকে। ছাপা হবার সময়ে ব্যারেলটি কাগজের সামনে খুব দ্রুত ঘোরে (সেকেণ্ডে 20 থেকে 21 পাক), এবং রোটারি প্রেসের মতো একটি ইঙ্ক (Ink Roller) রোলার ব্যারেলের গায়ে কালি মাখিয়ে দেয়। কাগজের অপর দিকে থাকে 120টি ব্যালিস্টিক প্রিন্ট হ্যামার (Ballistic Print Hammer)। এগুলি এমন ভাবে সাজানো থাকে

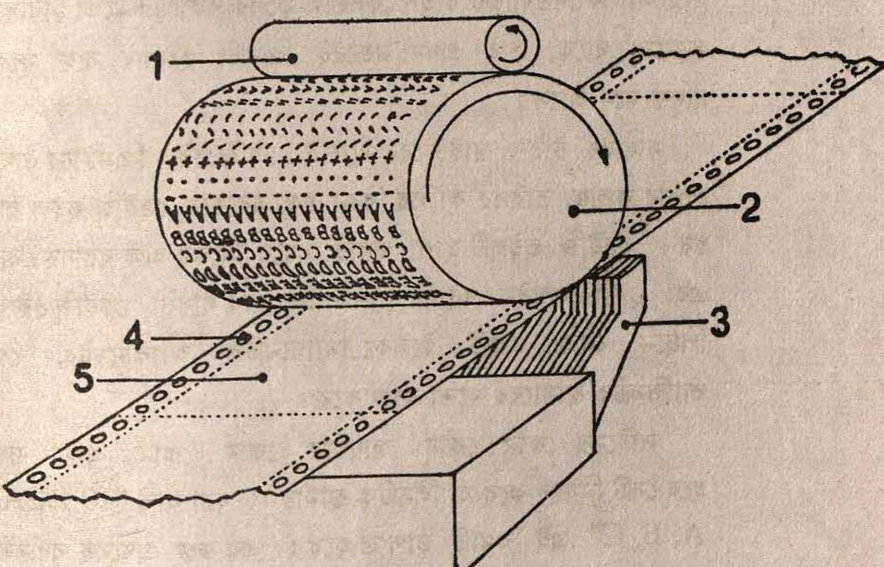
যাতে একটি হ্যামার ছিটকে কাগজে লাগলে সেটি কাগজকে প্রিন্ট ব্যারেলের গায়ে উঁচু হয়ে থাকা কোন একটি ক্যারেকটারের গায়ে চেপে ধরে। ব্যাপারটি টাইপ রাইটারের কার্য প্রণালীর ঠিক উল্টো, টাইপ রাইটারে চাবি টিপলে একটি হরফ বা সংখ্যা লেখা টাইপ ছিটকে উঠে কাগজের গায়ে প্রিনটিং রিবনকে চেপে ধরে। টাইপের চাপে রিবন থেকে ওই টাইপের কালির ছাপ কাগজে উঠে যায়। লাইন প্রিনটারের ক্ষেত্রে টাইপের গায়েই কালি লাগানো থাকে কিন্তু টাইপটি কাগজে এসে লাগেনা। ছাপ তোলার জন্য কাগজের পিছন থেকে একটি অত্যন্ত দ্রুত ব্যালিসটিক হ্যামার কাগজকে টাইপের গায়ে চেপে ধরে ছাপ তুলে নেয়।

লাইন প্রিন্টার

এবার দেখা যাক লাইন প্রিনটারে ছাপা এতো দ্রুত হয়ে কেন। ধরা যাক ব্যারেলটা ঘুরতে ঘুরতে কাগজের সামনে 120টি A লাগানো হরফের লাইন এলো। তখনই যে বাক্যটি ছাপতে হবে, তাতে যে যে জায়গায় যতগুলি A আছে, ঠিক সেই সেই জায়গার ব্যালিস্টিক হ্যামারগুলি ছিটকে গিয়ে কাগজে সেই লাইনের সমস্ত A হরফের ছাপ তুলে নেয়। এর পরই ব্যারেলটি ঘুরে 120 টি B হরফ লাগানো

চিত্র ৪৫

লাইন প্রিন্টারে ছাপার পদ্ধতি



1. ইংক রোলার
2. প্রিন্ট ব্যারেল
3. ব্যালিসটিক হ্যামার
4. স্প্রকেট হোল
5. ফ্যান ফোল্ড পেপার



ডেইজি হুইল প্রিন্টার

লাইন আসে। যে বাক্যটি ছাপতে হবে সেই বাক্যে যদি কোনো B হরফ থাকে, তবে সেই B হরফগুলি যে জায়গায় আছে, সে জায়গার হ্যামার ছিটকে গিয়ে কাগজে B-র ছাপ তুলে নেয়। এই ভাবে ব্যারেলটি যখন পুরো এক পাক ঘোরে, কাগজের ওপর 120টি অক্ষর দিয়ে গোটা লাইনই ছাপা হয়ে যায়। ব্যাপারটি এতো দ্রুত ঘটে যে মিনিটে 120 অক্ষর বিশিষ্ট 1250 লাইন লেখা ছেপে ফেলা এ ধরণের প্রিন্টারের পক্ষে মোটেই দুরূহ কাজ নয়। লাইন প্রিন্টারে প্রতি লাইন ছাপার পর ছাপার কাগজের ধারে যে গোল ফুটো বা স্প্রকেট (Sprocket) করা থাকে সেটির মধ্যে গলানো কাঁটা ওয়ালা চাকা, কাগজকে এক লাইন এগিয়ে দেয়। (এই ধরণের ছাপার কাগজকে কম্পিউটার পরিভাষায় বলা হয় ফ্যান ফোল্ড পেপার।

লাইন প্রিন্টারে ছাপার কাগজে প্রতি লাইন কোথায় ছাপা হবে সেটি নিখুঁত ভাবে নিয়ন্ত্রণ করা যায়। এই সুবিধার জন্য ছাপানোর ফর্মের বিভিন্ন খোপে লাইন প্রিনটারের সাহায্যে তথ্য ছাপানো যেতে পারে। তাই টেলিফোন বিল, ট্যাক্স বিল, ইলেকট্রিক বিল ইত্যাদি লাইন প্রিন্টারের সাহায্যে ছাপানো যায়।

ডেইজি হুইল প্রিন্টার

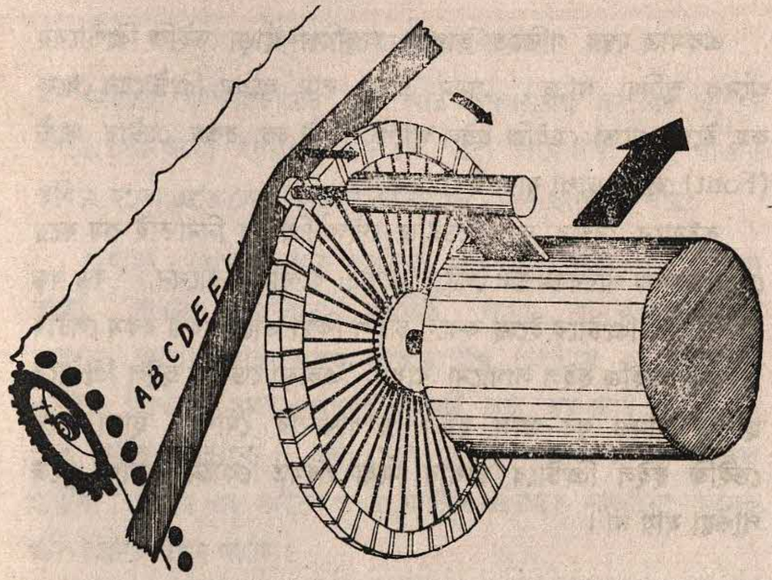
ডেইজি প্রিন্টারের মূল প্রিনটিং হেডটি ডেইজি (Daisy) ফুলের মতো দেখতে। একটি বৃত্তাকার রিং এর সঙ্গে নমনীয় সংযোগ ব্যবস্থার দ্বারা অ্যালফানিউমারিক টাইপ ফেসগুলি ফুলের পাপড়ির মতো চারদিকে ছড়ানো থাকে, তাই প্রথম নজরেই প্রিনটিং হুইলের সঙ্গে ফুলের সাদৃশ্য মনে আসে।

সাধারণ টাইপ রাইটারের মতোই ডেইজি হুইল প্রিনটারে প্রথমে ছাপার কাগজ, তারপর কালির রিবন এবং সবশেষে ডেইজি হুইল রাখা হয়। ডেইজি হুইলটি ছাপা হবার সময়ে নিজের অক্ষ বরাবর ঘোরে এবং ছাপার লাইন বরাবর পাশাপাশি সরে যায়। ডেইজি হুইলের পিছনে থাকে একটি ইলেকট্রোম্যাগনেটিক অ্যাকচুয়েটর, যেটি ব্যালিসটিক হ্যামারের মতো কাজ করে।

লাইনের কোন্ কোন্ জায়গায় কোন কোন্ অক্ষর ছাপা হবে সেটি নিয়ন্ত্রণ করে ব্যালিসটিক হ্যামার। ধরা যাক "10 READ A, B, C" এই কথাটি ছাপতে হবে। এর জন্য প্রথমেই ঘূর্ণায়মান ডেইজি হুইলটি কাগজের বাঁ দিকে সরে যাবে, তারপর ঘুরন্ত অবস্থাতে পুনরায় ডান দিকে ফিরে আসতে থাকবে। বাঁ দিকের

৭

ডেইজি হুইল প্রিন্টার



চিত্র ৪৬

ডেইজি হুইল প্রিন্টারে ছাপার পদ্ধতি দেখানো হয়েছে। ডেইজি হুইলটিকে তার অক্ষ বরাবর মটোরটি ঘোরাচ্ছে (মাঝের তীর চিহ্নের দিক বরাবর) এবং একই সঙ্গে মটোরটি ডান দিকের বড় তীর চিহ্নের দিকে সরে যাচ্ছে। মটোরের ওপরে বসানো ব্যালিসটিক হ্যামার সঠিক টাইপটি সামনে এলেই পিছন থেকে ধাক্কা দিয়ে কাগজের গায়ে (টাইপের মাধ্যমে) রিবনকে চেপে ধরে লেখা ছাপিয়ে দিচ্ছে।

মার্জিনের জায়গাটি বাদ দিয়ে যেই ডেইজি হুইলটি প্রথম ক্যারেকটার ছাপার জায়গায় আসবে তখনই একটি সংকেত চলে যাবে প্রিনটারের যান্ত্রিকব্যবস্থা নিয়ন্ত্রণের সার্কিটে। ওই অবস্থায় ঘূর্ণায়মান ডেইজি হুইলের 1 টাইপটি যেই প্রিনটিং পজিশনে আসবে অমনি ব্যালিস্টিক হ্যামার টাইপের পিছন থেকে ধাক্কা দিয়ে রিবনকে টাইপের সাহায্যে কাগজের ওপর চেপে ধরবে আর কাগজে 1 সংখ্যাটির ছাপ উঠে যাবে। এর পরই ডেইজি হুইলটি দ্বিতীয় ক্যারেকটার ছাপার জায়গায় সরে যাবে এবং নিজের অক্ষর বরাবর ঘুরে হুইলের 0 টাইপটি যেই প্রিনটিং পজিশনে আসবে অমনি ব্যালিস্টিক হ্যামার 0 টাইপকে ধাক্কা দেবে। এই ভাবে ক্রমশ বাঁ দিক থেকে ডান দিকে সম্পূর্ণ লেখাটি ছাপা হয়ে যাবে। ডেইজি প্রিন্টার খুব তাড়াতাড়ি ছাপতে পারে না। সেকেণ্ডে 40টির মতো ক্যারেকটার এতে ছাপা যায়। তাই খুব দ্রুত কাজ করে এমন কম্পিউটারের সঙ্গে ডেইজি প্রিন্টার লাগানো হয়না। প্রধানত মাইক্রো কম্প্যিউটার, পারশোনাল কম্পিউটার এবং কিছু কিছু মিনি কম্পিউটারের সঙ্গে ডেইজি হুইল প্রিন্টার লাগানো হয়।

একমাত্র মন্থর গতিতে ছাপার অসুবিধা ছাড়া ডেইজি প্রিন্টারের অনেক সুবিধা আছে। যেমন এদের দাম লাইন প্রিন্টারের থেকে কম, ইচ্ছে মতো ডেইজি হুইল পাল্টে পাল্টে বহু রকম লেটার ফন্টে (Font) লেখা ছাপা যায়, ইত্যাদি।

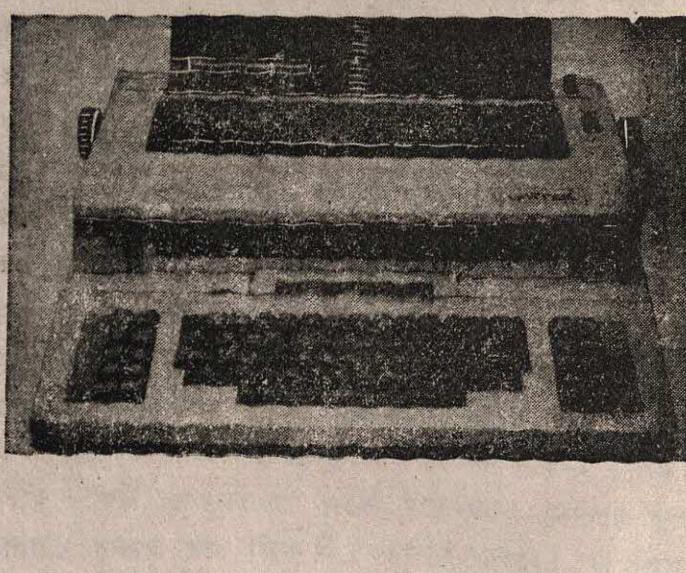
বর্তমানে বেশীর ভাগ ডেইজি হুইল প্রিন্টার নির্মাতাই কম করে গোটা দশেক পরিবর্তনশীল লেটার ফন্টের বন্দোবস্ত রাখেন। বড় বড় নির্মাতাদের প্রিন্টারে ইচ্ছে করলে চল্লিশ কিম্বা তারো বেশী রকম লেটার ফন্টযুক্ত ডেইজি হুইল লাগানো যায়। এছাড়া ডেইজি হুইল প্রিন্টারে ছাপার মানও খুব ভাল হয়। ছাপাখানার মেশিনে ছাপা আর ডেইজি হুইল প্রিন্টারে ছাপার মধ্যে গুণগত কোনো তফাৎই খুঁজে পাওয়া যায় না।

ইলেকট্রনিক টাইপ রাইটার

ইলেকট্রনিক টাইপ রাইটার

আধুনিক ইলেক্ট্রনিক টাইপ রাইটারকে কম্পিউটার প্রিন্টার হিসাবে ব্যবহার করা যায়। কম্পিউটারের সঙ্গে যুক্ত না করলে এগুলি সাধারণ টাইপ রাইটারের মতো কাজ করে। কিন্তু যেই এগুলিকে কম্পিউটারের I/O সকেটে লাগানো হয়, অমনি ইলেক্ট্রনিক টাইপ রাইটারটি একটি লেটার কোয়ালিটি ইম্প্যাক্ট প্রিন্টারে পরিবর্তিত হয়ে যায়। তবে এগুলি যেহেতু টাইপ রাইটারের কাজের জন্যই বিশেষ ভাবে তৈরী তাই সাধারণ ডেইজি হুইল প্রিন্টারের থেকে এদের দাম অনেক বেশী।

চিত্র ৪৭



ইলেকট্রনিক টাইপ রাইটার। এগুলিতেও ইম্প্যাক্ট প্রিনটিং ব্যবস্থা থাকে। বেশীর ভাগ ক্ষেত্রে লেখা ছাপার ব্যবস্থাটি ডেইজি হুইল কিম্বা গলফ্ বল প্রিন্টারের মত কাজ করে।

ইম্প্যাক্ট ম্যাট্রিক্স প্রিন্টার



ইম্প্যাক্ট ম্যাট্রিক্স প্রিন্টার

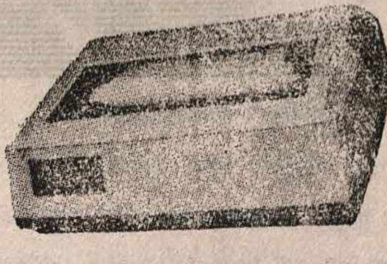
মাইক্রো কম্পিউটারের জন্য আর একধরনের জনপ্রিয় প্রিন্টার আছে, সেটি হলো ম্যাট্রিক্স প্রিন্টার। এরা দামে সস্তা এবং বাড়তি সুবিধে হলো এতে লেখা ছাড়াও গ্রাফিক্সের ছবিও ছাপানো যায়।

সাধারণ প্রিন্টার আর ম্যাট্রিক্স প্রিন্টারের মধ্যে মূল পার্থক্য হলো, সাধারণ প্রিন্টারে প্রতিটি সংখ্যা বা হরফের জন্য আলাদা আলাদা টাইপ থাকে, কিন্তু ম্যাট্রিক্স প্রিন্টারে তা থাকেনা। ম্যাট্রিক্স প্রিন্টারে সমস্ত ক্যারেক্টার সৃষ্টি করা হয় ছোট ছোট গোল গোল ফুটকির সাহায্যে। এই গোল ফুটকি বা Dot ছাপানোর জন্য ম্যাট্রিক্স প্রিন্টারে এক লাইনে অথবা ম্যাট্রিক্স হিসেবে সাজানো অনেক গুলি প্রিনটিং পিন থাকে।

ম্যাট্রিক্স প্রিন্টারে ছাপার পদ্ধতির সঙ্গে কম্পিউটার গ্রাফিক্সের অনেক মিল আছে। কম্পিউটার গ্রাফিক্সে যেমন VDU স্ক্রিনকে অজস্র পিক্সেলে ভেঙে নেওয়া হয় এবং ওই পিক্সেল গুলিকে ইচ্ছেমতো আলোকিত করে লেখা ফুটিয়ে তোলা হয়, তেমনি ম্যাট্রিক্স প্রিন্টারে সমস্ত লেখা এবং রেখা কালো ফুটকি সাজিয়ে গড়ে তোলা হয়। এই বইয়ে কম্পিউটার প্রিন্টারে ছাপার যে কটি নিদর্শন আছে, তার সবকটিই ম্যাট্রিক্স প্রিন্টারে ছাপানো। ইদানিং আমাদের দেশে এল. ই. ডি ম্যাট্রিক্স ডিসপ্লের চল হয়েছে। গোল গোল আলোর বিন্দু দিয়ে গড়ে তোলা একটি লম্বাটে ডিসপ্লে স্ক্রিনে বিভিন্ন কম্পানির বিজ্ঞাপন, লেখা, খবর ইত্যাদি ফুটে উঠে সচল অবস্থায় ডান দিক থেকে বাঁ দিকে সরে যায়। এখানে যে রকম প্রতিটি হরফ বা সংখ্যা, গোল আলোর বিন্দু দিয়ে তৈরী হয়, ম্যাট্রিক্স প্রিন্টারে ছাপা হরফ ও সংখ্যা গুলিও তেমনি কালো ফুটকি দিয়ে লেখা হয়।

চিত্র ৪৮

ইম্প্যাক্ট ম্যাট্রিক্স প্রিন্টার





ইম্প্যাক্ট ম্যাট্রিক্স প্রিণ্টার

বেশীর ভাগ ম্যাট্রিক্স প্রিণ্টারে অ্যালফানিউমারিক ক্যারেকটার লিখতে 7 × 5 ফুট্‌কি দিয়ে তৈরী একটি আয়তক্ষেত্রাকার জায়গায় ব্যবহার করা হয়। এই আয়তক্ষেত্রের মধ্যে যে 35টি ফুট্‌কি থাকে, তার মধ্যে কিছু কিছু জায়গায় ফুট্‌কি রেখে এবং বাকি জায়গায় ফুট্‌কি না রেখে সব ক্যারেকটারগুলি লেখা হয়।

লেখা ছাপানো জন্য অবশ্য 35 টি প্রিনটিং পিন লাগেনা। উল্লম্ব (Vertical) ভাবে সাজানো 7টি পিন হলেই চলে। এই 7 টি পিন পুরো প্রিনটিং লাইনে ওপর দিয়ে সরে সরে (Scan) যায় এবং যেখানে যেমন দরকার সেই মতো ফুট্‌কি প্রিণ্ট করে।

ম্যাট্রিক্স প্রিণ্টার, ডেইজি হুইলের থেকে দ্রুত প্রিণ্ট করতে পারে। তবে যেহেতু এর সমস্ত ক্যারকটার আলাদা আলাদা ফুটকি দিয়ে তৈরী, তাই এতে ছাপা বিষয় বস্তু দেখতে ভালো লাগেনা। সুন্দর ছাপার হরফ পেতে গেলে ডেইজি হুইলই আদর্শ। তবে ছবি বা গ্রাফ ছাপতে গেলে ম্যাট্রিক্স প্রিণ্টার ছাড়া গতি নেই। ইদানিং

চিত্র ৪৯



ম্যাট্রিক্স প্রিণ্টারে ছাপানো কম্পিউটার গ্রাফিক্সের নমুনা

অবশ্য কিছু ম্যাট্রিক্স প্রিন্টার তৈরী হচ্ছে যাতে 24 কিম্বা তারো বেশী প্রিনটিং পিন থাকে। এ ধরনের প্রিন্টারে খুব দ্রুত (300 ক্যারেকটার প্রতি সেকেণ্ডে) সাধারণ মানের ম্যাট্রিক্স প্রিনটিং করা যায়, কিম্বা ইচ্ছা করলে ছাপার গতি কমিয়ে (100 ক্যারেকটার প্রতি সেকেণ্ডে) ডেইজি হুইলের মতো সুন্দর ছাপার হরফে প্রিন্ট করা যায়।



থার্মাল প্রিন্টার

চিত্র ৫০

The **Macintosh**, on the other hand acts as a text entry cum editing terminal, although being a powerful computer by itself. Its ease-of-use and versatile capabilities takes it a generation ahead of others. It is capable of producing different styles like **BOLD**, *ITALICS*, **OUTLINE**, **SHADOW**, **UNDERLINE** and any *combinations* of these and sizes ranging from 4 Points to 127 Points, in the several ready typefaces offered. And whatever you see on the Macintosh's *bit mapped high resolution screen* **can be printed on the LaserWriter by just giving a print command through the mouse.**

থার্মাল ম্যাট্রিক্স প্রিন্টার

ম্যাট্রিক্স প্রিন্টারেই আর একটি বিশেষ রূপ হলো থার্মাল প্রিন্টার। একটু আগে যে ম্যাট্রিক্স প্রিন্টারের কথা আলোচনা করা হলো সে গুলি কাগজের গায়ে কালির রিবনকে প্রিনটিং পিন দিয়ে চেপে ছাপ তুলে ছাপার কাজ করে থাকে। আর থার্মাল প্রিন্টারে প্রিনটিং পিন বৈদ্যুতিক কারেন্ট পাঠিয়ে গরম করে রাখা হয়। ওই গরম পিনগুলি বিশেষ থার্মাল প্রিনটিং পেপারের গায়ে ঠেকালেই কাগজে কালো দাগ দেখা দেয় এবং ছাপার কাজ হয়ে যায়। এ ধরণের প্রিন্টারে কোনো কালি লাগেনা তবে কাগজটি বিশেষ থার্মাল পেপার হওয়ার জন্য লেখা ছাপার খরচ কিছু বেশী পড়ে।

দামের দিক থেকে থার্মাল প্রিন্টার বেশ সস্তা। যান্ত্রিক জটিলতা অন্যান্য প্রিন্টারের থেকে কম হওয়ায় এদের খুব ছোট আকারেও তৈরী করা যায়। আধুনিক পকেট পারশোনাল কম্পিউটারে (যেমন CASIO PB 300) এই প্রিন্টার ব্যবহার করা হয়।

ইলেকট্রোস্ট্যাটিক প্রিন্টার

ইলেক্ট্রোস্ট্যাটিক প্রিন্টারের কার্য প্রণালী অনেকটি প্লেন পেপার কপিয়ার (যেমন জেরক্স মেশিন) মেশিনের মতো। এই প্রিন্টারে কাগজের ওপর বৈদ্যুতিক চার্জের (Static Charge) সাহায্যে হরফগুলি

সর্বাধুনিক লেসার প্রিন্টারে ছাপার নমুনা। এ ধরনের প্রিন্টারে কালির বদলে লেসার রশ্মি ব্যবহার করা হয়। লেসার রশ্মিটি ফোটোগ্রাফিক ফিল্মে কিম্বা বিশেষ থার্মাল প্রিনটিং পেপারে অজস্র সূক্ষ্ম ডট্ বা বিন্দু ফুটিয়ে তোলে। সাধারণ চোখে সেই বিন্দু বা ডট্ ধরা পড়ে না।

বর্তমানে **PTS** (Photo Type Setting) পদ্ধতিতে বই বা পত্রিকা কম্পোজ করতে এই ধরনের প্রিন্টার ব্যবহার করা হচ্ছে।



ইংক জেট প্রিন্টার

লেখা হয়। চার্জের আকর্ষণে গুঁড়ো কালি কাগজের ওই সব অংশে লেগে যায়। তারপর উত্তাপ দিয়ে সেই গুঁড়ো কালিকে কাগজে এঁটে দেওয়া হয়।

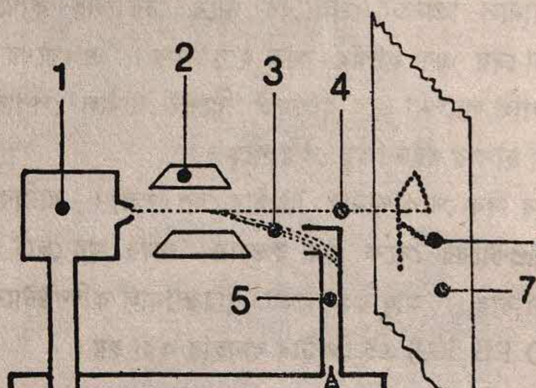
ইংক জেট প্রিন্টার

ইংক্ জেট (Ink Jet) প্রিন্টারে প্রথমে তরল কালিকে অ্যাটোমাইজার মেশিনের সাহায্যে অত্যন্ত ছোটো ছোটো বিন্দুতে বাষ্পের মতো ভেঙে নেওয়া হয়। এই বাষ্পের মতো কালির ফোয়ারাকে চার্জ করে ইলেক্ট্রোস্ট্যাটিক ডিফ্লেক্শন (Electrostatic Deflection) ব্যবস্থার সাহায্যে কাগজের ওপর ফুটিয়ে তুলতে ব্যবহার করা হয়।
(ঠিক যেমন ভাবে অসিলো স্কোপের ক্যাথোড রে টিউবে ইলেকট্রন বিমকে ইলেক্ট্রোস্ট্যাটিক ব্যবস্থার সাহায্যে নাড়িয়ে চাড়িয়ে ফস্ফরের ওপর রেখা ফুটিয়ে তোলা হয়)।

ইলেক্ট্রোস্ট্যাটিক এবং ইংক জেট, এ দুধরণের প্রিন্টারই দ্রুত লেখা ছাপাতে পারে। এদের তাই মিনি কম্পিউটার এবং মেইন ফ্রেম কম্পিউটারে বেশী ব্যবহার করা হয়।

চিত্র ৫১

ইংক জেট প্রিন্টারে লেখা ছাপানোর পদ্ধতি।



1. পাম্প এবং অ্যাটোমাইজার
2. ইলেকট্রোস্ট্যাটিক ডিফ্লেকশন প্লেট
3. বাড়তি কালি
4. ইংক্ জেট
5. বাড়তি কালির পাইপ
6. ছাপা হরফ
7. কাগজ

পরিচ্ছেদ ৮

বুলিয়ান অ্যালজেব্রা

বাইনারি নাম্বার সিসটেম আর তার যোগ বিয়োগ গুণ ভাগের নিয়ম আমরা জেনে গেছি। এবার জানা যাক কি ভাবে ইলেকট্রনিক সার্কিটের সাহায্যে এই বাইনারি যোগ-বিয়োগ ইত্যাদি করা হয়। আমরা বাইনারি যোগ বিয়োগের টেবিল দেখে কাগজে অঙ্ক কষে যোগফল বা বিয়োগ ফল বার করি। কিন্তু কম্পিউটারের সাহায্যে দুটো বাইনারি সংখ্যার যোগফল বার করতে হলে, সংখ্যা দুটির ডিজিটাল সংকেত একটি ইলেকট্রনিক সার্কিটে পাঠাতে হয় এবং সেই সার্কিটটি তৎক্ষণাৎ অঙ্ক কষে যোগফলের ডিজিটাল সংকেত বলে দেয়। পুরো ব্যাপারটি কিভাবে ইলেক্ট্রনিক সার্কিটের সাহায্যে করা হয় সেটি বোঝার জন্য আমাদের প্রথমে জানা দরকার বুলিয়ান অ্যালজেব্রা এবং সেই সঙ্গে ইলেক্ট্রনিক ডিজিটাল গেট সার্কিটগুলির সঙ্গেও পরিচয় লাভ করা দরকার। এই পরিচ্ছেদে আমরা বুলিয়ান অ্যালজেব্রা এবং পরবর্তী পরিচ্ছেদে ডিজিটাল গেটের সঙ্গে পরিচয় লাভ করবো।

বুলিয়ান অ্যালজেব্রা কিন্তু বেশ পুরোনো। 1854 সালে বিখ্যাত ইংরেজ গণিতজ্ঞ জর্জ বুল তাঁর গবেষনা গ্রন্থ 'An Investigation of the Laws of Thought, on which Are Founded the Mathematicial Theories of Logic And Probabilities'

৮

বুলিয়ান অ্যালজেব্রা

প্রকাশ করেন। সেই গ্রন্থ প্রকাশের সঙ্গে সঙ্গেই জন্ম নেয় এক নতুন প্রজন্মের অঙ্ক, যার নাম বুলিয়ান অ্যালজেব্রা। বুলিয়ান অ্যালজেব্রা মূলত লজিকের সত্যি অথবা মিথ্যে এই দুটি স্তরের ওপর ভিত্তি করে তৈরী হয়েছে। বুলিয়ান অ্যালজেব্রার এই দ্বিস্তরীয় (Two State) বৈশিষ্ট্যের জন্য, পরবর্তীকালে যখন কম্পিউটারে বাইনারি নাম্বার সিসটেম ব্যবহার শুধু হলো, তখন বুলিয়ান অ্যালজেব্রার সত্যি এবং মিথ্যেকে বাইনারি 1 এবং 0 দিয়ে পরিবর্তন করে নিতেই কম্পিউটারের অঙ্ক কষার সমস্ত গানিতিক সমস্যা বুলিয়ান অ্যালজেব্রার সাহায্য সমাধান করা সম্ভব হলো। মজার কথা হলো, জর্জ বুল যখন তাঁর বুলিয়ান অ্যালজেব্রা আবিষ্কার করেন, তখনও কিন্তু তাঁর অ্যালজেব্রার সূত্র কম্পিউটারের অঙ্ক কষার কাজে ব্যবহার করার কথা ভাবা-হয়নি। পরবর্তী কালে যখন ডিজিটাল কম্পিউটারের প্রচলন হলো, তখনই দেখা গেলো কম্পিউটারের গাণিতিক কাজ কর্ম ব্যাখ্যা করতে বুলিয়ান অ্যালজেব্রা কি ভীষণ দরকারী।

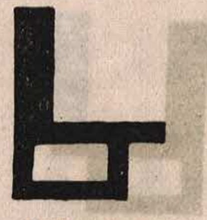
কম্পিউটার বিজ্ঞান পড়তে গেলে বুলিয়ান অ্যালজেব্রা জানা অতি অবশ্য প্রয়োজনীয়। বর্তমানে স্কুল লেভেলেও বুলিয়ান অ্যালজেব্রা এবং সেট থিওরী পড়ানো হচ্ছে। জটিলতার দিক থেকে দেখতে গেলে বুলিয়ান অ্যালজেব্রা কিন্তু খুব সহজ। তবে এর তাৎপর্য এবং উপযোগীতাকে উপলব্ধি না করতে পারলে অনেক সময়ই পুরো ব্যাপারটি অদ্ভুত, এবং কোনো কোনো ক্ষেত্রে কঠিন বলে মনে হয়। আর যদি বুঝতে চেষ্টা করা হয় কি ভাবে এই অ্যালজেব্রার নিয়মাবলিকে ইলেকট্রনিক সার্কিটের সাহায্যে নকল করে কম্পিউটারের মধ্যে সমস্ত হিসেব নিকেষ করা হচ্ছে, তখনই বিষয়টি দারুন সহজ আর আকর্ষণীয় হয়ে ওঠে।

বুলিয়ান গাণিতিক সূত্র

আমরা স্কুলে যে অ্যালজেব্রা শিখেছি তার সঙ্গে বুলিয়ান অ্যালজেব্রার প্রধান পার্থক্য হলো—স্কুলের অ্যালজেব্রায় যে কোনো অজানা পরিবর্তনশীল বস্তুর (Variable) মান যে কোনো গানিতিক সংখ্যা হতে পারে। যেমন আমাদের পরিচিত অ্যালজেব্রায় $X+Y=Z$ সমিকরণে X Y এবং Z-এর মান যে কোনো ধণাত্মক, ঋণাত্মক কিম্বা ভগ্নাংশ সংখ্যা হতে পারে।

কিন্তু বুলিয়ান অ্যালজেব্রায় একটি ভেরিয়েবলের এর কেবল

মাত্র দুটোই মান হতে পারে, "সত্যি" অথবা "মিথ্যা" যাকে বাইনারি 1 কিম্বা 0 দিয়ে বোঝানো যায়।



OR অপারেশন

সুতরাং $X+Y=Z$ ধরনের কোনো সমিকরণ বুলিয়ান অ্যালজেব্রায় লেখা হলে সেখানে X, Y এবং Z প্রতিটির কেবল দুটোই মান হতে পারে 0 এবং 1। এই মানগুলি বসিয়ে বুলিয়ান অ্যালজেব্রার যোগের নিয়ম মেনে কেবল চারটে সামিকরণ লেখা যায়, সে গুলি হলো :—

$$0+0=0\cdots\cdots(1)$$
$$0+1=1\cdots\cdots(2)$$
$$1+0=1\cdots\cdots(3)$$
$$1+1=1\cdots\cdots(4)$$

প্রথম 1, 2 এবং 3 নম্বর সমিকরণটি আমাদের পরিচিত অ্যালজেব্রার যোগের নিয়মের সঙ্গে মিলে যাচ্ছে, কিন্তু 4 নম্বর সমিকরণ $1+1=1$-এর সঙ্গে পরিচিত অ্যালজেব্রার যোগের কোনো মিলই নেই। সুতরাং বোঝাই যাচ্ছে বুলিয়ান যোগের + চিহ্ন সাধারণ অ্যালজেব্রার যোগকে বোঝায় না। বুলিয়ান যোগকে বলা হয় লজিকাল অ্যাডিশন (Logical Addition) অথবা লজিকাল অর অপারেশন (Logical OR Operation)। যাতে সাধারণ + চিহ্নের সঙ্গে OR চিহ্ন গুলিয়ে না যায়, তার জন্য বুলিয়ান অ্যালজেব্রায় অনেক সময় লজিকাল OR-কে + চিহ্নের বদলে U চিহ্নের মাধ্যমে লেখা হয়।

$$0U0=0$$
$$0U1=1$$
$$1U0=1$$
$$1U1=1$$

কম্পিউটার টেকনোলজিতে অবশ্য + চিহ্নটিই বেশী ব্যবহার করা হয়। পড়ার সময় + কিম্বা U চিহ্নের জায়গায় অর্ বলে পড়া উচিত, যেমন, $A+B=C$ যদি বুলিয়ান অ্যালজেব্রায় লেখা হয়, তবে সেটি পড়তে হবে এ অর বি ইজইকুয়াল টু সি (A OR B is equal to C) হিসেবে।

বুলিয়ান অ্যালজেব্রায় শুধু মাত্র বুলিয়ান যোগ এবং গুণ-এর মাধ্যমে সমস্ত অঙ্ক করা হয়। আমরা বুলিয়ান যোগের অর্থাৎ OR অপারেশনের নিয়ম শিখেছি, এবার শিখবো গুণের নিয়ম! বুলিয়ান

৮

AND অপারেশন

গুণকে লজিকাল মাল্টিপ্লিকেশন বা লজিকাল অ্যাণ্ড অপারেশন (Logical Multiplication, Logical AND Operation) বলা হয়। এ বিষয়ে নিয়মটি হলো ঃ—

$$
\begin{array}{|l|}
\hline
0 \cdot 0 = 0 \cdots\cdots (1) \\
0 \cdot 1 = 0 \cdots\cdots (2) \\
1 \cdot 0 = 0 \cdots\cdots (3) \\
1 \cdot 1 = 1 \cdots\cdots (4) \\
\hline
\end{array}
$$

এই চারটি সমিকরণের মাধ্যমেই সমস্ত রকম বুলিয়ান গুণ করা হয়। পড়বার সময়ে A.B=C কে পড়তে হবে এ অ্যাণ্ড বি ইজ ইকুয়াল টু সি।

বুলিয়ান কমপ্লিমেণ্ট

বুলিয়ান ভেরিয়েবলের দুটি সম্ভাব্য মান 0 এবং 1কে একটি অপরটির কমপ্লিমেণ্ট বলা হয়। অর্থাৎ 1 এর কমপ্লিমেণ্ট হলো 0 এবং 0-র কমপ্লিমেণ্ট হলো 1।

এই কথাটিকে গানিতিক ভাবে লেখা হয় এইভাবে যেমন A-র কমপ্লিমেণ্ট হলো $\overline{A}$

যদি A-র মান হয় 0 তবে $\overline{A}=1$

এবং যদি A-র মান হয় 1 তবে $\overline{A}=0$

কমপ্লিমেণ্ট ব্যাপারটি খুবই গুরুত্বপূর্ণ। বুলিয়ান অ্যালজেব্রার অঙ্ক কষতে গেলে প্রায়ই কমপ্লিমেণ্ট নির্ণয় করার প্রয়োজন হয়।

৮

বুলিয়ান ফর্মুলা

এক নজরে বুলিয়ান যোগ, গুণ ও ইনভার্সানের মূল ফর্মুলা

(1) $0+0=0$

(2) $0+1=1$

(3) $1+0=1$

(4) $1+1=1$

(5) $0 \cdot 0=0$

(6) $0 \cdot 1=0$

(7) $0 \cdot 0=0$

(8) $1 \cdot 1=1$

(9) $\overline{1} = 0$

(10) $\overline{0} = 1$

মূল ফর্মুলা থেকে প্রমাণিত অন্যান্য গুরুত্বপূর্ণ ফর্মুলা

(1) $O+A=A$

(2) $1+A=1$

(3) $A+A=A$

(4) $A+\overline{A}=1$

(5) $A \, . \, \overline{A}=0$

(6) $O \, . \, \overline{A}=0$

(7) $1 \, . \, A=A$

(8) $A.A=A$

(9) $\overline{\overline{A}} = A$

(10) $A+B=B+A$

(11) $A \, . \, B=B \, . \, A$

(12) $A+(B+C)=(A+B)+C$

(13) $A.(B.C)=(A.B).C$

(14) $A.(B+C)=A.B+A.C$

(15) $A+A.B=A$

(16) $A.(A+B)=A$

(17) $(A+B).(A+C)=A+B.C$

(18) $A+\overline{A}.B=A+B$

(19) $A.B+B.C+\overline{B}.C=A.B+C$

৮

বুলিয়ান ফর্মূলা

বুলিয়ান অঙ্ক কষার নিয়ম

যে পদ্ধতিতে বুলিয়ান অঙ্ক সমাধান করা হয়, তাকে বলা হয় পারফেক্ট ইণ্ডাক্শন (Perfect Induction)। এই পদ্ধতিতে যে ফর্মূলা প্রমাণ করতে হবে, তার বাঁদিক এবং ডানদিকের ভেরিয়েবল গুলির সকল সম্ভাব্য মান একটি তালিকায় লেখা হয় (তালিকাটির নাম ট্রুথ টেবিল)। যদি ভেরিয়েবল গুলির সকল মানের ক্ষেত্রেই বাঁদিক এবং ডানদিকের মান মিলে যায়, তাহলে প্রমাণিত হয় ওই ফর্মূলাটি সঠিক। এই পদ্ধতিতে আগের পৃষ্ঠার (16) নম্বর ফর্মূলাটি প্রমাণ করা হলো।

A	B	A+B	A.(A+B)	A
0	0	0	0	0
1	0	1	1	1
0	1	1	0	0
1	1	1	1	1

ওপরের ট্রুথ টেবিল (Truth Table) থেকে দেখা যাচ্ছে ভেরিয়েবল A এবং B-র যে কোনো মানের জন্য সর্বদা A.(A+B) এবং A-র মান অভিন্ন। সুতরাং Perfect Induction অনুযায়ী ফর্মূলাটি সঠিক।

ডি মরগ্যানের সূত্র

একই পদ্ধতিতে আরো দুটি গুরুত্বপূর্ণ বুলিয়ান সূত্র প্রমান করা যায়। সূত্র দুটি বুলিয়ান অঙ্কের জন্য অত্যন্ত প্রয়োজনীয় এবং এরা ডি মরগ্যানের সূত্র (De Morgan's Theorem) নামে পরিচিত। নিচে সূত্র দুটি দেওয়া হলো এবং সেই সঙ্গে সূত্র দুটি প্রমাণ করা হলো।

$$(\overline{A+B})=\bar{A}.\bar{B} \quad \cdots \quad \cdots \quad (1)$$

$$(\overline{A.B})=\bar{A}+\bar{B} \quad \cdots \quad \cdots \quad (2)$$

প্রমাণ ঃ—

A	$\bar{A}$	B	$\bar{B}$	A+B	A.B	$\overline{A+B}$	$\bar{A}.\bar{B}$	$\overline{A.B}$	$\bar{A}+\bar{B}$
0	1	0	1	0	0	1	1	1	1
1	0	0	1	1	0	0	0	1	1
0	1	1	0	1	0	0	0	1	1
1	0	1	0	1	1	0	0	0	0

এই তালিকা থেকে দেখা যাচ্ছে A, B-র সকল সাম্ভাব্য মানের জন্য $\overline{A+B}=\bar{A}.\bar{B}$ এবং $\overline{A.B}=\bar{A}+\bar{B}$ সুতরাং ডি মরগ্যানের সূত্র প্রমাণিত হলো।

পরিচ্ছেদ ৯

ডিজিটাল গেট

যে সমস্ত ইলেকট্রনিক সার্কিটের সাহায্যে (বুলিয়ান সূত্র অনুযায়ী) বাইনারি অঙ্ক হয়, তাদের নাম ডিজিটাল গেট (Digital Gate) সার্কিট। প্রথম প্রজন্মের কম্পিউটারে এই গেটগুলি রিলে (Relay) যন্ত্রের সাহায্যে তৈরী করা হতো। আধুনিক আই সি প্রযুক্তিতে সব রকম ডিজিটাল গেট, আই সি হিসেবে তৈরী করা হয়। যেমন বুলিয়ান OR অপারেশন করার জন্য ডিজিটাল OR গেট আই সি পাওয়া যায়। আবার বুলিয়ান AND, NOT ইত্যাদি অপারেশনের জন্য পাওয়া যায় ডিজিটাল AND, NOT গেট আই সি। এছাড়া এই তিনটি প্রধান গেটের সমন্বয়ে বিভিন্ন বুলিয়ান ফর্মূলা অনুসারে কাজ করে এমন ডিজিটাল গেটও তৈরী করা হয়, যেমন NOT এবং AND এর সমন্বয়ে NAND গেট, NOT এবং NOT-এর সমন্বয়ে Buffer গেট ইত্যাদি।

ডিজিটাল OR গেট

যে কোন ডিজিটাল গেটই হলো একটি ইলেকট্রনিক সার্কিট, যার এক বা একাধিক ইনপুট টার্মিনাল (Terminal) এবং একটি আউটপুট টার্মিনাল (Terminal) থাকে। ইনপুট টার্মিনালে/



OR গেট

টার্মিনালগুলিতে ডিজিটাল 0 (0 ভোল্ট বিদ্যুৎ বিভব) কিম্বা 1 (2 ভোল্ট বিদ্যুৎ বিভব) সংকেত পাঠালে, সেই সার্কিট কোন একটি বিশেষ বুলিয়ান সূত্র অনুসারে আউটপুট টার্মিনালে তথ্য (0 ভোল্ট কিম্বা 2 ভোল্ট) আউটপুট দেয়।

একটি ডিজিটাল OR গেটের কথাই ধরা যাক, যাতে দুটি ইনপুট টার্মিনাল (A, B) এবং একটি আউটপুট টার্মিনাল (C) আছে। এই OR গেটের A এবং B টার্মিনালে সংকেত পাঠালে গেটটি বুলিয়ান OR অপারেশনের সূত্র অনুযায়ী C টার্মিনালে ডিজিটাল আউটপুট দেয়। নানা রকম ইনপুটে OR গেটের কি কি ডিজিটাল আউটপুট পাওয়া যায় তা নিচের সারণীতে দেখানো হল। এই সারণীকে

OR গেটের ইনপুট গুলিকে সমান্তরালে (Parallel) যুক্ত ইলেকট্রিক সুইচ হিসেবে কল্পনা করা যায়। এভাবে সংযুক্ত সুইচের মধ্যে যে কোন একটি অন্ করলেই সার্কিট অন্ হয়। আর সার্কিটকে অফ্ করতে হলে সব কটি সুইচই অফ্ করতে হয়। ঠিক তেমনি OR গেটের সব কটি ইনপুট যতক্ষণ না 0 ভোল্ট (অফ্) করা হয়, আউটপুটটি ততক্ষণ 2 ভোল্ট (অন্) হয়ে থাকে।

ইনপুট		আউটপুট
A	B	C
0 ভোল্ট	0 ভোল্ট	0 ভোল্ট
2 ভোল্ট	0 ভোল্ট	2 ভোল্ট
0 ভোল্ট	2 ভোল্ট	2 ভোল্ট
2 ভোল্ট	2 ভোল্ট	2 ভোল্ট

ভোল্টের মানে না লিখে ডিজিটাল 1 এবং 0 সংকেত হিসেবে লিখলে দাঁড়ায় এই রকম—

ইনপুট		আউটপুট
A	B	C
0	0	0
1	0	1
0	1	1
1	1	1

একে বলা হয় OR গেটের ট্রুথ টেবিল (Truth Table)। ট্রুথ টেবিল বা সত্যের সারণী নামটি বেশ মজার, তাই না? নামটি এসেছে বুলিয়ান অ্যালজেব্রা থেকে। বুলিয়ান অ্যালজেব্রায় এই

ধরণের সারণী বিভিন্ন অঙ্ক সমাধান করতে ব্যবহার করা হয়। তবে বুলিয়ান সারণীতে ডিজিটাল 1 এর জায়গায় True (সত্য) এবং ডিজিটাল 0-র জায়গায় False (মিথ্যা) শব্দ ব্যবহার করা হয়। সুতরাং সেই ভাবে লিখলে ডিজিটাল OR গেটের Truth Table-এর চেহারা হয় এই রকম—

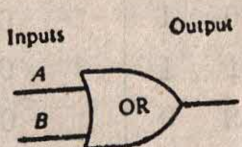


AND গেট

ইনপুট		আউটপুট
A	B	C
False	False	False
True	False	True
False	True	True
True	True	True

ডিজিটাল OR গেটে যে কেবলমাত্র দুটিই ইনপুট থাকবে এমন কোনো বাধ্যবাধকতা নেই। তিন, চার, পাঁচ ইনপুট বিশিষ্ট OR, গেটও পাওয়া যায়।

ইলেকট্রনিক সার্কিট আঁকার সময়ে বিভিন্ন ডিজিটাল গেটগুলিকে আলাদা আলাদা সাংকেতিক চিহ্নের সাহায্যে প্রকাশ করা হয়। ছবিতে ডিজিটাল OR গেটের বহুল ব্যবহৃত সাংকেতিক চিহ্ন দেখানো হল।



চিত্র ৫২

OR গেটের সাংকেতিক চিহ্ন

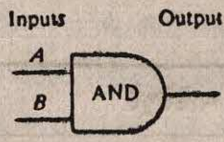
ডিজিটাল AND গেট

কম্পিউটারের অভ্যন্তরে বুলিয়ান AND অপারেশন করা হয় ডিজিটাল AND গেটের সাহায্যে। AND গেটের সাংকেতিক চিহ্ন



AND গেট

এবং ট্রুথ টেবিল নিচে দেওয়া হলো। OR গেটের মতই AND গেটেও দুই এর বেশী ইনপুট থাকতে পারে।



ইনপুট AND		আউটপুট
A	B	C
0	0	0
1	0	0
0	1	0
1	1	**1**

চিত্র ৫৩

AND গেটের সাংকেতিক চিহ্ন এবং Truth Table

AND গেটের সঙ্গে শ্রেণী সমবায়ে (Series) সংযুক্ত ইলেট্রিক সুইচের তুলনা করা হয়। শ্রেণী সমবায়ে যুক্ত সুইচের মধ্যে সব কটি যতক্ষণ না অন্ করা হচ্ছে, ততক্ষণ সার্কিটটি পূর্ণ (অন্) হয় না। ঠিক তেমনি AND গেটের সব কটি ইনপুটে ডিজিটাল 1 সংকেত না দিলে আউটপুটে কখনো ডিজিটাল 1 পাওয়া যায় না।

তিন ইনপুট বিশিষ্ট AND গেটের Truth Table. এই Truth Table থেকে পরিষ্কার বোঝা যাচ্ছে AND গেটের সব কটি ইনপুটে ডিজিটাল 1 দিলে তবেই আউটপুটে ডিজিটাল 1 পাওয়া যায়।

ইনপুট AND			আউটপুট
A	B	C	D
0	0	0	0
0	1	0	0
0	0	1	0
0	1	1	0
1	0	0	0
1	1	0	0
1	0	1	0
1	1	1	1

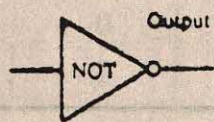


NOT গেট
NOR গেট

ডিজিটাল NOT গেট

বুলিয়ান ইনভার্সন বা কমপ্লিমেন্ট করার জন্য কম্পিউটারে ব্যবহার করা হয় NOT গেট। এই গেটটিতে কেবলমাত্র একটি ইনপুট এবং একটি আউটপুট টার্মিনাল থাকে। ইনপুটে যদি 0 ভোল্ট অর্থাৎ ডিজিটাল 0 সংকেত দেওয়া হয় তবে আউটপুটে 2 ভোল্ট অর্থাৎ ডিজিটাল 1 সংকেত পাওয়া যায়। আবার যখন ইনপুটে 2 ভোল্ট দেওয়া হয়, তখন আউটপুটে 0 ভোল্ট পাওয়া যায়। সুতরাং একটি NOT গেটের ইনপুটে যে ডিজিটাল তথ্যই দেওয়া হোক না কেন আউটপুটে সর্বদা কমপ্লিমেন্ট বা ইনভার্টেড তথ্যই পাওয়া যায়। তাই অনেক সময় NOT গেটকে ইনভার্টারও (Inverter) বলা হয়ে থাকে। নিচে NOT গেটের সাংকেতিক চিহ্ন এবং ট্রুথ টেবিল দেওয়া হলো।

ইনপুট	আউটপুট
1	0
0	1



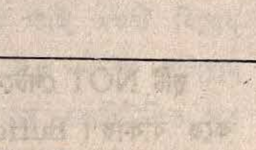
চিত্র ৫৪

ডিজিটাল NOR গেট

একটি সাধারণ OR গেটের আউটপুটে যদি একটি NOT গেট জুড়ে দেওয়া হয়, তবে যা পাওয়া যায়, সেটি হলো NOR গেট। আউটপুট টার্মিনালে NOT গেট লাগানোর দরুণ একটি OR গেটের কোনো বিশেষ ইনপুটের ক্ষেত্রে যা আউটপুট হয়, NOR গেটের ক্ষেত্রে ঠিক তার উল্টো বা কমপ্লিমেন্ট আউটপুট পাওয়া যায়। নিচে NOR গেটের ট্রুথ টেবিল দেওয়া হল।

চিত্র ৫৫

ইনপুট	NOR	আউটপুট
A	B	C
0	0	1
1	0	0
0	1	0
1	1	0



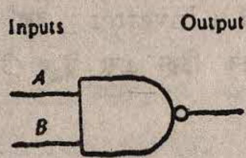
ডিজিটাল NAND গেট



NAND গেট

NOR গেটের মতই একটি সাধারণ AND গেটের আউটপুটে NOT গেট লাগিয়ে NAND গেট তৈরী করা হয়। ফলে বিভিন্ন পরিস্থিতিতে একটি সাধারণ AND গেট যা আউটপুট দিত তার ঠিক কমপ্লিমেন্ট আউপুট পাওয়া যায় একটি NAND গেট থেকে। নিচে NAND গেটের সাংকেতিক চিহ্ন এবং ট্রুথ টেবিল দেওয়া হলো। OR, AND বা NOR এর মতই NAND গেটেও দুই-এর বেশী ইনপুট টার্মিনাল থাকতে পারে।





ইনপুট NAND		আউটপুট
A	B	C
0	0	1
1	0	1
0	1	1

চিত্র ৫৬

বাঁদিকে TTL প্রযুক্তিতে নির্মিত NAND গেট IC দেখা যাচ্ছে। ডানদিকে NAND গেটের সাংকেতিক চিহ্ন দেখানো হয়েছে।

Buffer গেট

দুটি NOT গেটকে পরপর শ্রেণী সমবায়ে (Series) সংযুক্ত করে বাফার (Buffer) গেট তৈরী করা হয়। NOT গেটদুটি এই ভাবে লাগানোর ফলে ইনপুটের তথ্য পরপর দুবার কমপ্লিমেন্ট হয়ে আউটপুটে পুনরায় ইনপুটের তথ্যই ফিরে আসে। অর্থাৎ বাফার গেটের ইনপুটে যা তথ্য দেওয়া হয়, সেই একই তথ্য আউটপুটে পাওয়া যায়। কম্পিউটারে বাফারের কাজ হলো বিভিন্ন পেরিফেরালসের মধ্যে বৈদ্যুতিক সংযোগ বা ইন্টারফেস (Interface) ঘটানো। নিচের সারণীতে Buffer-এর ট্রুথ টেবিল দেওয়া হলো।

ইনপুট	আউটপুট
1	1
0	0

ডিজিটাল ExOR গেট



ExOR গেট

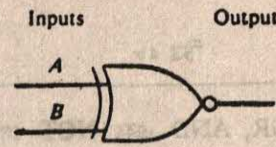
কম্পিউটারের ডিজিটাল সার্কিট নির্মাণে একটি বিশেষ বুলিয়ান ফর্মুলা খুব বেশী ব্যবহার করতে হয়। ফর্মুলাটি বুলিয়ান অ্যালজেব্রার নিয়মে লিখলে দাঁড়ায় এই রকম :—

$$C = \overline{\bar{A}.\bar{B} + A.B} \quad \cdots \quad \cdots \quad (a)$$

ওপরের ফর্মুলায় A এবং B হলো ইনপুট এবং C হলো আউটপুট। ফর্মুলাটির Truth Table নিচে দেওয়া হলো।

ইনপুট		ExOR গেটের Truth Table					আউটপুট
A	B	$\bar{A}$	$\bar{B}$	A.B	$\bar{A}.\bar{B}$	$\bar{A}.\bar{B}+A.B$	$\overline{\bar{A}.\bar{B}+A.B}$
0	0	1	1	0	1	1	0
1	0	0	1	0	0	0	1
0	1	1	0	0	0	0	1
1	1	0	0	1	0	1	0

ExOR গেটের আউটপুটে NOT গেট লাগিয়ে ExNOR গেট তৈরী



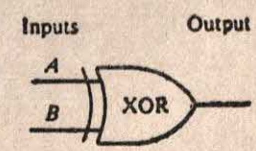
করা হয়। ছবিতে ExNOR গেটের সাংকেতিক চিহ্ন দেখানো হলো।

ডিজিটাল সার্কিট নির্মাণ সহজতর করার জন্য তাই একটি বিশেষ ইলেকট্রনিক গেট সার্কিট তৈরী করা হয়, যেটির ট্রুথ টেবিল বুলিয়ান ফর্মুলা (a) মেনে চলে। এই বিশেষ গেটের নাম এক্সক্লিউসিভ অর গেট (Exclusive OR Gate) বা সংক্ষেপে ExOR গেট। ExOR গেটের ট্রুথ টেবিলের সঙ্গে বুলিয়ান ফর্মুলা (a)-এর ট্রুথ টেবিল মিলিয়ে দেখুন, দেখবেন দুটি অবিকল একই রকম। বাস্তবে ExOR

চিত্র ৫৭

ইনপুট	(ExOR)	আউটপুট
A	B	C
0	0	0
1	0	1
0	1	1
1	1	0

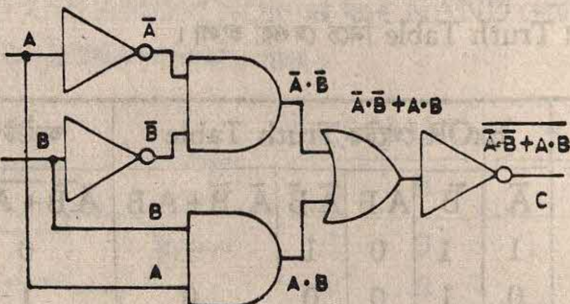
ExOR-এর সাংকেতিক চিহ্ন এবং ট্রুথ টেবিল।





হাফ অ্যাডার

গেটগুলি OR, AND কিম্বা NOT গেটের মত দেখতে I C প্যাকেজে পাওয়া যায়। IC-র অভ্যন্তরে কি ভাবে OR, AND এবং NOT গেট ব্যবহার করে ExOR গেট তৈরী করা হয় সেটি ছবিতে দেখানো হল। একটু খেয়াল করে লক্ষ্য করলেই বুঝতে পারবেন অবিকল বুলিয়ান ফর্মুলা (a) অনুসারে গেট গুলি সংযুক্ত করে ExOR গেটটি তৈরী করা হয়েছে।



চিত্র ৫৮

OR, AND এবং NOT গেট ব্যবহার করে ExOR গেট নির্মাণ প্রণালী।

বাইনারি হাফ অ্যাডার সার্কিট

কম্পিউটার বিজ্ঞানে বাইনারি যোগ সব থেকে গুরুত্বপূর্ণ গাণিতিক অপারেশন, কারণ প্রায় সমস্ত মাইক্রো কম্পিউটারেই যাবতীয় গাণিতিক কাজকর্ম বাইনারি যোগের মাধ্যমে করা হয়। আমরা তো জানি, যে গুণ হলো বারম্বার যোগ করার সংক্ষিপ্ত পদ্ধতি। তাই যে কোন গুণই বার বার যোগ করে করা যায়। তেমনি বাইনারি বিয়োগও কমপ্লিমেন্ট বার করে বাইনারি যোগের পদ্ধতি অনুসারে করা যায়। বিয়োগ করতে পারলেই ভাগ করাও সম্ভব হয়, কারণ গুণের মত ভাগ হলো বারম্বার বিয়োগ করার সংক্ষিপ্ত পদ্ধতি। গণিতে এই মূল চারটি অপারেশন (যোগ, বিয়োগ, গুণ, ভাগ) করা গেলে যে কোনো অন্যান্য গাণিতিক অপারেশন, যেমন, বর্গফল, বর্গমূল ইত্যাদি করাও সম্ভব হয়। কম্পিউটার অসম্ভব দ্রুত গতিতে যোগ করতে পারে বলে খুব বড় বড় অঙ্ক এই ভাবে কষতে খুবই অল্প সময় লাগে।

এবার আসুন, দেখা যাক বাইনারি যোগ করার Adder সার্কিট কি ভাবে ডিজিটাল লজিক গেট জুড়ে জুড়ে তৈরী করা হয়। এটি

বোঝার আগে অবশ্য আর একবার বাইনারি যোগের সারণীতে চোখ বুলিয়ে নেওয়া দরকার।

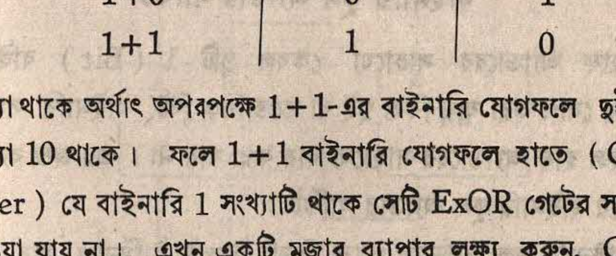
এই সারণী থেকে পরিষ্কার দেখা যাচ্ছে, বাইনারি যোগফলের প্রথম বিট সংখ্যাগুলি ExOR গেটের ট্রুথ টেবিলের সঙ্গে অবিকল মিলে গেছে। তবে ExOR-এর আউটপুটে সর্বদা 1 বিট্ (Bit) বাইনারি



হাফ অ্যাডার

বাইনারি যোগ (A+B)	দ্বিতীয় বিট্	প্রথম বিট্
0+0	0	0
0+1	0	1
1+0	0	1
1+1	1	0

সংখ্যা থাকে অর্থাৎ অপরপক্ষে 1+1-এর বাইনারি যোগফলে দুই বিট্ সংখ্যা 10 থাকে। ফলে 1+1 বাইনারি যোগফলে হাতে (Carry Over) যে বাইনারি 1 সংখ্যাটি থাকে সেটি ExOR গেটের সাহায্যে পাওয়া যায় না। এখন একটি মজার ব্যাপার লক্ষ্য করুন, Carry Over বা হাতে যে সংখ্যাটি থাকে সেটির ট্রুথ টেবিল AND গেটের ট্রুথ টেবিলের সঙ্গে অবিকল মিলে গেছে। সুতরাং একটি ExOR এবং AND গেটের ইনপুট দুটি যদি পরস্পরের সঙ্গে (ছবিতে যে ভাবে দেখানো আছে সে ভাবে) যুক্ত করা হয় তবে ইনপুটে দুটি বাইনারি



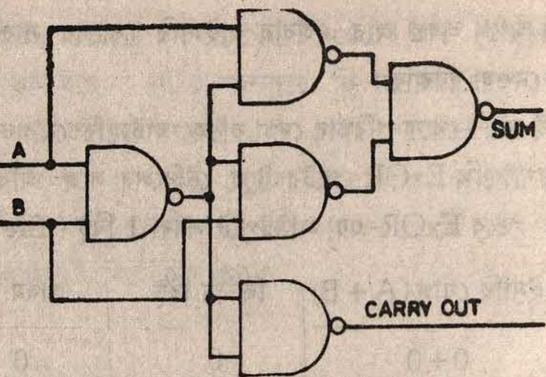
চিত্র ৫৯

বাঁদিকে ExOR এবং AND গেট দিয়ে তৈরী হাফ অ্যাডার সার্কিট। ডানদিকে হাফ অ্যাডারের সাংকেতিক চিহ্ন।

সংখ্যার ডিজিটাল তথ্য দিলেই ExOR গেটের আউটপুটে যোগফলের একক অবস্থানের উত্তর এবং AND গেটের আউপুটে Carry Over এর তথ্য পাওয়া যায়। এই ভাবে তৈরী হয়ে যায় একটি বাইনারি যোগ করার ইলেকট্রনিক সার্কিট, বাইনারি হাফ অ্যাডার (Half Adder)।



ফুল অ্যাডার



চিত্র ৬০

কেবল মাত্র NAND গেট ব্যবহার করে তৈরী হাফ অ্যাডার সার্কিট।

বাইনারি ফুল অ্যাডার সার্কিট

হাফ অ্যাডারের সাহায্যে কেবল দুটি 1 (Bit) বাইনারি সংখ্যার যোগ করা যায় কিন্তু দুই বা ততোধিক বিট্ বাইনারি সংখ্যার যোগফল হাফ অ্যাডারের সাহায্যে বার করা যায় না। এর জন্য ব্যবহার করা হয় বাইনারি ফুল অ্যাডার সার্কিট।

বাইনারি ফুল অ্যাডারে থাকে তিনটি ইনপুট টার্মিনাল, যার মধ্যে দুটি টার্মিনাল A এবং B-তে যে বাইনারি সংখ্যা দুটি যোগ করতে হবে তার ডিজিটাল সংকেত পাঠানো হয়। A এবং B ছাড়া তৃতীয় ইনপুটটির নাম ক্যারি ইন্ (Carry In) ইনপুট। তিনটি ইনপুট ছাড়া, ফুল অ্যাডারের দুটি আউটপুট থাকে। একটি Sum এবং অপরটি Carry out বা হাতে কতো রইলো তার আউটপুট।

সারণী

ফুল অ্যাডারের Truth Table

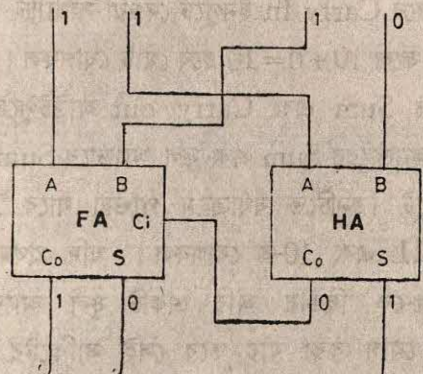
A—ইনপুট
B—ইনপুট
Ci—ক্যারি ইন ইনপুট
Co—ক্যারি আউট আউটপুট
S—সাম আউটপুট

ইনপুট			আউটপুট	
A	B	Ci	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

ধরা যাক আমরা ছুটি ছুই বিট্‌, বাইনারি সংখ্যা যোগ করবো। সংখ্যা ছুটি হলো 11 এবং 10। এই যোগফল হাফ অ্যাডারের সাহায্যে বার করা যাবে না, কারণ হাফ অ্যাডারে মোটে ছুটি ইনপুট টার্মিনাল আছে। এক্ষেত্রে আমাদের ব্যবহার করতে হবে একটি হাফ অ্যাডার এবং একটি ফুল অ্যাডার। ছবিতে যে ভাবে দেখানো হয়েছে সে ভাবে অ্যাডার ছুটি লাগাতে হবে। হাফ অ্যাডারের ইনপুটে 11 এবং 10-র প্রথম অবস্থানের (Least significant digit) সংখ্যা ছুটি (11-এর ক্ষেত্রে ডান দিকের 1 সংখ্যাটি এবং 10-র ক্ষেত্রে ডানদিকের 0 সংখ্যাটি) যুক্ত হবে। হাফ অ্যাডার সংখ্যা ছুটি যোগ করে Sum আউটপুটে 1+0=1 আউটপুট দেবে। হাতে কিছু না



ফুল অ্যাডার

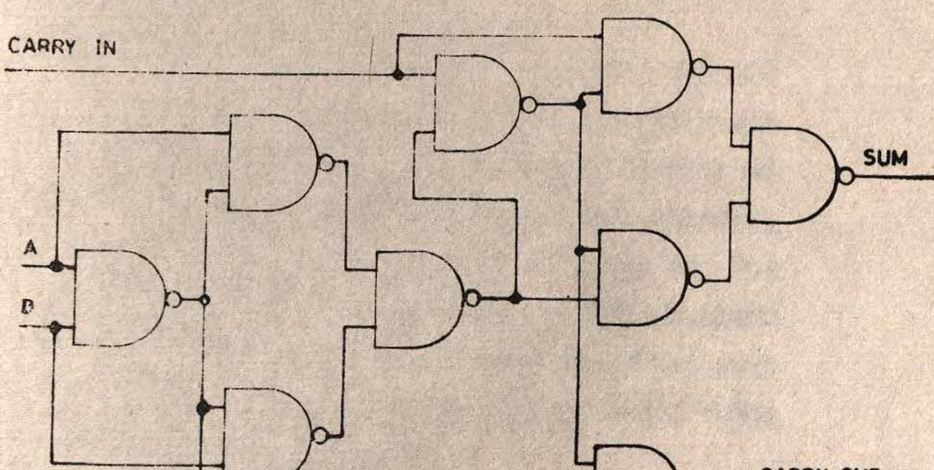


চিত্র ৬১

ছুটি 2 Bit সংখ্যা যোগ করার সার্কিট

চিত্র ৬২

ডিজিটাল NAND গেট ব্যবহার করে তৈরী ফুল অ্যাডার সার্কিট।





ফুল অ্যাডার

থাকার জন্য হাফ অ্যাডারের Carry out আউটপুটে 0 তথ্য পাওয়া যাবে। এই আউটপুট ফুল অ্যাডারের Carry In ইনপুটে লাগানো আছে। ফলে Carry In ইনপুটে 0 তথ্য আসবে। এছাড়া ফুল অ্যাডরের ইনপুট ছুটিতে যুক্ত হবে 11 এবং 10-এর দ্বিতীয় অবস্থানের (Most significant digit) সংখ্যা ছুটি (11-এর ক্ষেত্রে বাঁ দিকের 1 এবং 10-এর ক্ষেত্রে বাঁ দিকের 1 সংখ্যাটি)। ফুল অ্যাডার প্রথমে 1+1=10 যোগফল বার করবে। তারপর তার সঙ্গে যোগ করবে Carry In ইনপুটে দেওয়া সংখ্যাটি (এক্ষেত্রে সেটি হলো 0)। ফলে 10+0=10 হবে মোট যোগফল। এই যোগফল ফুল অ্যাডারের Sum এবং Carry out আউটপুটে পাওয়া যাবে। সুতরাং হাফ অ্যাডারের Sum এবং ফুল অ্যাডারে Sum এবং Carry out আউটপুট তিনটিতে যথাক্রমে পাওয়া যাবে 1, 0, 1 তথ্য। এটাই হলো 11 এবং 10-র যোগফল। যদি প্রথম ফুল অ্যাডারের Carry out-কে দ্বিতীয় আর একটি ফুল অ্যাডারের Carry In-এর সঙ্গে যোগ করা যায় তবে সেই সার্কিটের সাহায্যে আমরা ছুটি 3 বিট্ সংখ্যার যোগফল বার করতে পারবো। কার্যক্ষেত্রে কিন্তু আমাদের একটি একটি করে ফুল অ্যাডার জুড়ে বাইনারি যোগ করার সার্কিট তৈরী করতে হয় না। সাধারণত এক সঙ্গে 4 কিম্বা ততোধিক বিট্ বাইনারি সংখ্যা যোগ করার সার্কিট IC হিসেবেই কিনতে পাওয়া যায়।

পরিচ্ছেদ ১০

সফট্ওয়্যার এবং অপারেটিং সিসটেম

পৃথিবীর সমস্ত যন্ত্রই তৈরী করা হয় কোনো একটি বিশেষ কাজের উপযোগী করে। কিন্তু এর একমাত্র ব্যতিক্রম কম্পিউটার। কম্পিউটার যন্ত্রকে সেই দৃষ্টিভঙ্গি থেকে বিচার করলে বলতে হয় সম্পূর্ণ অকেজো ইলেকট্রনিক সার্কিটের একটি জঞ্জাল বিশেষ। কিছু মাত্র অত্যুক্তি নয়, খোদ কম্পিউটার পরিভাষাতেই কম্পিউটারের সমস্ত যন্ত্রপাতিকে হার্ডওয়্যার (Hardware) অর্থাৎ লোহালক্কড় বলে অভিহিত করা হয়! কারণ, কেবলমাত্র যন্ত্রপাতিগুলি এক সঙ্গে সংযুক্ত করে সুইচ্ অন্ করলে কম্পিউটারটি নিজে থেকে কিছুই করতে পারে না। কম্পিউটারকে দিয়ে সামান্যতম কোনো কাজ করাতে হলেই দরকার পড়ে কম্পিউটার প্রোগ্রাম।

প্রোগ্রামই কম্পিউটার হার্ডওয়্যারে প্রাণ প্রতিষ্ঠা করে এবং অপ্রয়োজনীয় ইলেকট্রনিক জঞ্জাল স্বরূপ যন্ত্রটিকে দিয়ে পৃথিবীর প্রায় সব রকম সম্ভাব্য কাজ করিয়ে নেয়। কম্পিউটার পরিভাষায় এই নির্দেশকে বলা হয় সফট্ওয়্যার (Software)। অথচ মজার ব্যাপার দেখুন, সফট্ওয়্যার কিন্তু কোনো যান্ত্রিক পার্টস বা সংযুক্তি নয়, শ্রেফ প্রোগ্রামারের মাথা খাটিয়ে বার করা বিভিন্ন কাজের উপযোগী প্রোগ্রাম বা নির্দেশের সমষ্টি। অনেকটা রান্নার বইতে লেখা বিভিন্ন রান্নার প্রণালীর মতো। কম্পিউটার প্রোগ্রামার হতে গেলে তাই কম্পিউটার কলকব্জা সম্বন্ধে জ্ঞান আবশ্যক নয়।

১৩

মনিটর প্রোগ্রাম

এমন অনেক প্রোগ্রামার আছেন, যাঁরা আদৌ বিজ্ঞানের ছাত্র নন, এবং হার্ডওয়্যার সম্বন্ধে কিছুই জানেন না। অথচ সফট্ওয়্যার রচনার দক্ষতাতে তাঁরা যে কোনো হার্ডওয়্যার পারদর্শী বিজ্ঞানের ছাত্রকে হারিয়ে দিতে পারেন! এ যেন অনেকটা মোটর গাড়ির মতো ব্যাপার। ভালো গাড়ি চালাতে জানার সঙ্গে যেমন গাড়ির কলকব্জা সম্বন্ধে জ্ঞানের কোনো সম্বন্ধ নেই, সেই রকমই সফট্ওয়্যার রচনার জন্য হার্ডওয়্যারের জ্ঞান নিতান্ত অপ্রয়োজনীয় (অবশ্য মেশিন ল্যাংগোয়েজ এবং অ্যাসেমব্লি ল্যাঙ্গোয়েজ এর ব্যতিক্রম)।

কম্পিউটার পরিভাষায় সফট্ওয়্যার বলতে যে কোনো রকম স্তর ও পর্যায়ের প্রোগ্রাম বোঝায়। তাই প্রোগ্রামারের লেখা প্রোগ্রামকেও সফট্ওয়্যার বলা হয়, আবার কম্পিউটার কোম্পানি থেকে ROM এ ভরে দেওয়া ফার্মওয়্যারকেও সফট্ওয়্যার বলা হয়। কার্যগত দিক থেকে কম্পিউটার সফট্ওয়্যারকে মোট তিন ভাগে ভাগ করা যায়।

(1) মনিটর প্রোগ্রাম (Monitor Program) এবং অপারেটিং সিস্টেম (Operating system)।

(2) ইউটিলিটি সফট্ওয়্যার (Utility software)।

(3) অ্যাপ্লিকেশন প্রোগ্রাম (Application program)।

মনিটর প্রোগ্রাম

মনে করা যাক আমরা ইলেকট্রনিক RAM, ROM, C-P-U, I/O, এবং অন্যান্য ডিজিটাল IC জুড়ে জুড়ে একটি সম্পূর্ণ কম্পিউটার তৈরী করে ফেললাম। সব কিছু মিলিয়ে দেখা গেল নির্ভুল হয়েছে। এবার মেইনস্ সকেটে প্লাগ গুঁজে সুইচ্ অন্ করা হলো। কিন্তু কি আশ্চর্য, কম্পিউটারতো কোনোই কাজ করছে না! ফের মিলিয়ে দেখা হলো, সবই ঠিক আছে। তাহলে? সকেটে কোনো গণ্ডগোল আছে নাকি? নিশ্চিত হওয়ার জন্য দোকান থেকে কেনা একটি কম্পিউটার ওই সকেটে লাগানো হলো। সেটি কিন্তু দেখা গেলো ঠিকই চলছে।

সব হার্ডওয়্যার তো ঠিক আছে, তবে গণ্ডগোলটা কোথায়? এবার মনে পড়লো, তাইতো, আমাদের কম্পিউটার ROM এ কোনোই প্রোগ্রাম ভরে রাখা হয়নি। তাই কম্পিউটার চালু করার পর কি

করতে হবে সেই নির্দেশ না পেয়ে C.P.U. কাজ থামিয়ে বসে আছে! এমনকি আমরা বাইরে থেকে প্রোগ্রাম করতে চাইলেও, সেই প্রোগ্রাম কি করে পড়তে হবে না বুঝতে পেরে C.P.U কিছুই করতে পারছে না। এই অচল কম্পিউটার সচল করতে আমাদের ROM-এ একটি বিশেষ সফট্ওয়্যার লিখে দিতে হবে, যেটি কম্পিউটার অন্ করার পরই C.P.U.-কে কি কি কাজ করতে হবে বলে দেবে, আর সেই সঙ্গে C.P.U-এর বিভিন্ন কাজ কর্মকেও নিয়ন্ত্রণ করবে। এই বিশেষ সফট্ওয়্যারকেই বলা হয় মনিটর সফট্ওয়্যার বা মনিটর প্রোগ্রাম।

মনিটর সফট্ওয়্যার কম্পিউটার নির্মাতাই ROM-এ লিপিবদ্ধ করে কম্পিউটারে বসিয়ে দেন। ফলে মনিটর সফট্ওয়্যারটি কম্পিউটার ব্যবহারকারী বা প্রোগ্রামারের কাছে অদৃশ্য থাকে। কিন্তু চোখের আড়ালে থেকেও এটি সর্বদা কম্পিউটার ব্যবহারকারী বা প্রোগ্রামারের লেখা প্রোগ্রামের সঙ্গে C. P. U.-কে সংযুক্ত করে রাখে। মনিটর প্রোগ্রামের সাহায্য ছাড়া কম্পিউটার-প্রোগ্রামার C. P. U.-এর সঙ্গে সংযোগ স্থাপন করতে পারেন না। বস্তুত এটিই হলো ন্যূনতম সফট্ওয়্যার, যেটি ব্যবহার করে কোনো কম্পিউটার হার্ডওয়্যারকে ব্যবহারোপযোগী করে তোলা যায়।

অপারেটিং সিস্টেম

কম্পিউটার প্রোগ্রামিংকে আরো সহজতর করে তোলার জন্য ইদানিং সমস্ত কম্পিউটারে মনিটর প্রোগ্রামের সঙ্গে আরো একটি বাড়তি সফট্ওয়্যার সংযুক্ত করা হয়। একে বলে অপারেটিং সিস্টেম (Operating system) সফট্ওয়্যার।

অপারেটিং সিস্টেম বা সংক্ষেপে OS সফট্ওয়্যারটি কম্পিউটারকে করে তোলে আরো বেশী 'জীবন্ত' যাকে কম্পিউটার পরিভাষায় বলা হয় User Friendly বা বন্ধুভাবাপন্ন। ইউজার ফ্রেণ্ডলি বলতে বোঝায় এমন একটি ব্যবস্থা যেটি কম্পিউটার ব্যবহারকারীর পরিশ্রম অনেকাংশে লাঘব করে দেয় এবং সেই সঙ্গে মনিটর স্ক্রিনের মাধ্যমে ব্যবহারকারীর সঙ্গে কথোপকথন চালায়। এই বিশেষ বৈশিষ্ট্যের ফলে ব্যবহারকারীর মনে হয় কম্পিউটারটি কেবলমাত্র যন্ত্রই নয়, তার বুঝি জীবন্ত মানুষের মতো চিন্তা ভাবনা করার ক্ষমতা আছে।

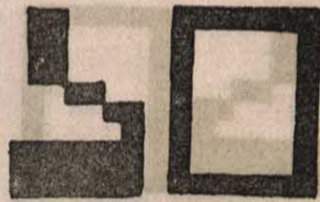
অপারেটিং সিস্টেম

উদাহরণ স্বরূপ বলা যায় কম্পিউটারের OS এর সাহায্যে আমরা অনায়াসে এমন ব্যবস্থা করতে পারি যাতে কম্পিউটার অন্ করলেই স্ক্রিনে যে সাইন অন্ মেসেজ (Sign on message) ফুটে ওঠে, তাতে লেখা হয়ে যায় 'Hello', কিম্বা এমন ব্যবস্থাও করা যায় যাতে বিশেষ পরিস্থিতিতে প্রোগ্রামারকে সাহায্য করার জন্য মনিটর স্ক্রিনে উপযুক্ত নির্দেশ ফুটে ওঠে, ইত্যাদি ইত্যাদি। এছাড়া কম্পিউটার প্রোগ্রামে যে সব গাণিতিক পদ্ধতি বেশী ব্যবহার করা হয় (যেমন বর্গমূল, বর্গফল ইত্যাদি ইত্যাদি), সেই সব গাণিতিক পদ্ধতি বা ফর্মুলা OS এর সাহায্যে আগে থেকেই কম্পিউটার ROM এ লিখে রাখা যায়। ফলে প্রোগ্রাম লেখার সময় প্রোগ্রামারকে ওই সমস্ত পদ্ধতির পুরো ফর্মুলা লিখতে হয় না, কেবল পদ্ধতিটির সংকেত উল্লেখ করলেই হয়।

শুধু তাই নয়, মনিটার সফট্ওয়ারের সাহায্যে কম্পিউটারকে একমাত্র মেশিন ল্যাংগোয়েজে (বাইনারি অথবা হেক্স সংখ্যার সাহায্যে) প্রোগ্রাম করা যায়। কিন্তু OS ব্যবহার করে BASIC, FORTRAN, COBOL প্রভৃতি হাই লেভেল প্রোগ্রামিং ভাষায় কম্পিউটারকে প্রোগ্রাম করা চলে। OS-এর অন্তর্গত Interpreter কিম্বা Compiler সফট্ওয়্যার হাই লেভেল প্রোগ্রামকে মেশিন ভাষায় অনুবাদ করে দেয়।

তাছাড়া কম্পিউটারের মাধ্যমে চিঠি, রচনা ইত্যাদি টাইপ করার সময়, কিম্বা লিখিত প্রোগ্রামের ভুল ত্রুটি সংশোধনের জন্য টাইপরাইটারের মতো ব্যবস্থা দরকার পড়ে। এই ব্যবস্থা শুধু মাত্র হার্ডওয়্যারের সাহায্যে করা যায় না। এটিও করে দেয় ওই OS সফট্ওয়্যার। OS-এর এই বিশেষ ব্যবস্থাকে অনেক সময় পৃথক ভাবে এডিটর প্রোগ্রাম (Editor pragram) বলা হয়। এডিটর প্রোগ্রামে কম্পিউটারকে বলে দেওয়া হয় কোন্ চাবি টিপলে দুটি হরফের মধ্যে এক ঘর করে ফাঁক রাখতে হবে, কোন্ চাবি টিপলে একটি হরফের স্থানে অন্য হরফ পাল্টে বসাতে হবে ইত্যাদি। প্রোগ্রামিং এর সময় প্রোগ্রামারকে এই সব কাজের প্রোগ্রাম নিয়ে চিন্তা করতে হয় না। কী বোর্ডের বিশেষ বিশেষ চাবি টিপলেই সমস্ত কাজ হয়ে যায়। কিন্তু পুরো কাজটাই প্রোগ্রামারের অজান্তে করিয়ে নেয় কম্পিউটরের OS সফট্ওয়্যার।

কম্পিউটার থেকে সেকেণ্ডারি মেমরি বা বাড়তি স্মৃতি ভাণ্ডারে তথ্য লেখা কিম্বা কোনো একটি বিশেষ তথ্য খুজে এনে দেওয়ার কাজও করে দেয় OS সফট্ওয়্যার। অর্থাৎ এক কথায় কম্পিউটার' সহজে পরিচালনার জন্য প্রায় সমস্ত রকম কাজ করার নির্দেশই লিপিবদ্ধ করা থাকে OS সফট্ওয়্যারে।



অপারেটিং সিস্টেম

1950 সালের আগে অবশ্য OS-এর প্রচলন ছিল না। সর্বপ্রথম আমেরিকার জেনর‍্ল্ মটোর রিসার্চ ল্যাবরেটারি IBM কোম্পানির জন্য OS তৈরী করেন। আরো পরে 1976 সাল নাগাদ পারশোনাল কম্পিউটার (Personal Computer) জগতে OS-এর প্রচলন শুরু হয়। বর্তমানে প্রায় সব কম্পিউটারেই OS ব্যবহার করা হচ্ছে।

ছোট মাইক্রোকম্পিউটার নির্মাতারা তাঁদের কম্পিউটারে মনের মতো সুযোগ সুবিধার ব্যবস্থা করতে আলাদা আলাদা OS ব্যবহার করেন। ফলে কোনো দুটি পৃথক কোম্পানির কম্পিউটারে কখনোই একই ধরনের প্রোগ্রামের সুযোগ সুবিধা পাওয়া যায় না। কোনো কোম্পানির কম্পিউটারে যদি এডিটর প্রোগ্রামটি ভাল হয়, তো অন্য কোম্পানির কম্পিউটারে গ্রাফিক্স আঁকতে সাহায্যকারী সফট্ওয়্যারটি হয় আরো বেশী আকর্ষণীয়। এতে সব থেকে বেশী অসুবিধা হয় প্রোগ্রামারদের। কারণ OS-এর পরিবর্তনে কেবলমাত্র প্রোগ্রাম করার সুযোগ সুবিধাই নয়, কম্পিউটার প্রোগ্রামের ভাষাতেও বিস্তর প্রভেদ সৃষ্টি হয়। এর ফলে কোনো একটি কোম্পানির কম্পিউটারকে যে ভাষায় প্রোগ্রাম করলে সেটি কাজ করবে, ঠিক সেই প্রোগ্রামটিই আবার অন্য কোম্পানির কম্পিউটারে সম্পূর্ণ অচল হয়ে যায়। কম্পিউটার পরিভাষায় এই ভাষাগত প্রভেদকে বলা হয় ডায়ালেক্ট (Dialect)।

বড় কম্পিউটারে ডায়ালেক্টের ঝামেলা এড়াতে কোম্পানি নির্বিশেষে কেবল কতগুলি বিশেষ OS ব্যবহার করা হয়। ফলে বিভিন্ন কোম্পানির একই OS যুক্ত কম্পিউটারে একই প্রোগ্রাম বিনা পরিবর্তনে চালানো যায়।

১০

কম্পিউটার ক্রাইম

কম্পিউটার ক্রাইম

নামটি শুনেই কেমন খট্‌কা লাগছে তাই না? আসলে উন্নত দেশগুলিতে অপরাধও উন্নত হয়ে থাকে। আমরা কাগজে গাড়ি চুরির ঘটনা পড়ি, আর অ্যামেরিকায় প্রাইভেট প্লেন চুরি হয়। লস্‌ অ্যাঞ্জেলেসে প্রাইভেট প্লেন মালিকেরা তাঁদের প্লেন হ্যাঙারে রাখার আগে প্রপেলারে শেকল বেঁধে তালা লাগিয়ে দেন। নইলে কে জানে কখন কোন্‌ পাইলট চোর ওই প্লেন নিয়ে চম্পট্‌ দেবে!

কম্পিউটার ক্রাইম ব্যাপারটি আরো সাংঘাতিক। বর্তমানে উন্নত দেশগুলিতে বেশীর ভাগ তথ্য সংরক্ষণ করা হয় কম্পিউটারের মাধ্যমে। যেমন প্রতিরক্ষা, জনসংযোগ, ব্যাংক, পুলিশ ক্রাইম রেকর্ড, সরকারি তথ্য দপ্তর ইত্যাদি ইত্যাদি। এই সব তথ্য অত্যন্ত গোপনীয় এবং ভুল লোকের হাতে পড়লে সাংঘাতিক বিপদজনকও বটে। তথ্যগুলি যাতে গোপন থাকে তার সুরক্ষা ব্যবস্থা করা হয় OS সফট্‌ওয়্যারের সাহায্যে। ফলে কম্পিউটার প্রোগ্রামাররা স্বাভাবিক ভাবে কী বোর্ডের মাধ্যমে হাজার চেষ্টা করলেও কম্পিউটার স্মৃতি ভাণ্ডার থেকে কোনো তথ্য গোপনে পড়ে নিতে পারেন না কিম্বা ভুল তথ্য ইচ্ছাকৃত ভাবে ঢুকিয়েও দিতে পারেন না।

কিন্তু যদি সর্ষের মধ্যেই ভূত থাকে? অর্থাৎ OS সফট্‌ওয়্যারেই যদি ইচ্ছাকৃত ভাবে এমন নির্দেশ ঢুকিয়ে দেওয়া যায় যাতে কোনো একটি সাংকেতিক বাক্য টাইপ করলে তথ্য ভাণ্ডারে ইচ্ছে মতো তথ্য লেখা বা পড়ার সুরক্ষা ব্যবস্থাটি সাময়িক ভাবে অকেজো হয়ে যায়? হ্যাঁ এটা করা মোটেই অসম্ভব নয়! কোনো ভালো কম্পিউটার প্রোগ্রামার মাথা খাটিয়ে এমন ব্যবস্থা অনায়াসেই করে ফেলতে পারেন।

এখন প্রশ্ন উঠতে পারে, মনিটর এবং OS সফট্‌ওয়্যার তো ROM-এ লিপিবদ্ধ থাকে। এর তথ্য কি করে পরিবর্তন করা যাবে? উত্তর হলো ছোট মাইক্রোকম্পিউটার ছাড়া একটু বড় আর দামী কম্পিউটারে OS সফট্‌ওয়্যারটি ROM-এ লেখা থাকে না। এটি একটি ফ্লপি ডিস্কে রেকর্ড করা থাকে। প্রোগ্রামার যখন প্রোগ্রাম করেন, তখন তাঁর প্রোগ্রামের সঙ্গে ওই OS সফট্‌ওয়্যারকে RAM-এ ঢুকিয়ে দেওয়া হয়। অর্থাৎ প্রতিবার প্রোগ্রামের সময় ফ্লপি

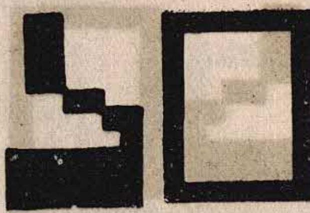
থেকে OS-কে RAM-এ ভরে দিতে হয়। আর ফ্লপিতে রেকর্ড করা OS-কে পরিবর্তন করা মোটেই দুরূহ ব্যাপার নয়।

এতো কেবল মাত্র একটি পদ্ধতি। OS-এর পরিবর্তন ছাড়াও স্মৃতি ভাণ্ডারে তথ্য চুরি করার আরো অনেক উপায় আছে!

শুধু তথ্য চুরিই নয়, ইচ্ছাকৃত ভাবে কম্পিউটারের কাজ কর্মে বিঘ্ন ঘটানোর প্রচেষ্টাও কম্পিউটার ক্রাইম বলেই গণ্য করা হয়। আগে যুদ্ধের সময় কলকারখানা ধ্বংস করতে কতই কসরৎ করতে হত, লক্ষ লক্ষ টাকার মারণাস্ত্র, প্লেন, মিসাইল ব্যবহার করেও অনেক সময় অভিষ্ট সিদ্ধি হত না। কিন্তু আধুনিক কম্পিউটার নিয়ন্ত্রিত অস্ত্রশস্ত্র বিকল করতে সামান্য স্ক্রু ড্রাইভারও দরকার নেই, শুধু OS-এ কিছু ইচ্ছাকৃত ভুল ত্রুটি ঢুকিয়ে দাও, ব্যাস কাজ শেষ।

এ ধরণের প্রচেষ্টা মোটেই গল্প কথা নয়, ইতিপূর্বে বহুবার বিদেশে এই ঘটনা ঘটেছে এবং এখনো এই ঘটনার ক্ষুদ্র সংস্করণ অবিরত ঘটে যাচ্ছে বিদেশের স্কুল কলেজগুলিতে। ছাত্ররা কম্পিউটার নিয়ে কাজ করার সময় অনেক ক্ষেত্রে খেলার ছলে OS-এর তথ্য অদল বদল করে দেয় এবং এই তুচ্ছ ঘটনা থেকেই উৎপত্তি হয় ভবিষ্যতে বড় বড় কম্পিউটার ক্রাইম সংঘটিত করার পরিকল্পনা।

তবে কম্পিউটার সফট্ওয়্যার ডিজাইনাররাও থেমে নেই। তাঁরাও অবিরত OS-গুলিকে আরো বেশি নিরাপদ করে তোলার প্রচেষ্টা চালিয়ে যাচ্ছেন। কিন্তু সম্পূর্ণ ভাবে নিরাপদ OS এখনও তৈরী করা সম্ভব হয়নি।



অ্যাপ্লিকেশন প্রোগ্রাম

অ্যাপ্লিকেশন প্রোগ্রাম

কম্পিউটারকে দিয়ে কোনো বিশেষ কাজ করিয়ে নেওয়ার জন্য প্রোগ্রামার বাইরে থেকে যে প্রোগ্রাম কম্পিউটারে ইনপুট করেন, তাকে বলে অ্যাপ্লিকেশন প্রোগ্রাম (Application Program)। কম্পিউটার শিক্ষা প্রতিষ্ঠানগুলি শিক্ষার্থীদের এই অ্যাপ্লিকেশন প্রোগ্রাম লিখতেই শেখান। পরবর্তী পরিচ্ছেদে আমরাও শিখবো কি করে অ্যাপ্লিকেশন প্রোগ্রাম লিখতে হয়।

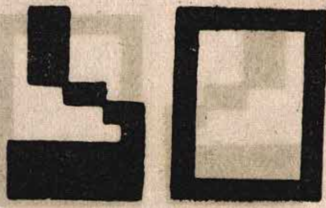
ছোটো ছোটো অ্যাপ্লিকেশন প্রোগ্রাম লেখা ক্যালকুলেটর চালানোর মতই সহজ। কিন্তু বড় কাজের অ্যাপ্লিকেশন প্রোগ্রাম লেখা অত্যন্ত

কঠিন কাজ। যেমন ধরুন, ইন্টার কন্টিনেন্টাল ব্যালিসটিক মিশাইল বা ICBM-এর গতিপথ নিয়ন্ত্রণ করার প্রোগ্রাম কিম্বা, মহাশূন্যে রকেট নিয়ন্ত্রণের প্রোগ্রাম ইত্যাদি। এ ধরনের এক একটি প্রোগ্রাম তৈরী করতে এক সঙ্গে বহু সুশিক্ষিত প্রোগ্রামার এবং বিজ্ঞানীর দু' তিন বছর সময় লেগে যায়। অতদূরও যাবার দরকার নেই, ভিডিও গেম পার্লারে পয়সা ফেলে যে স্পেশ ইনভেডর, নাইট ড্রাইভার, প্যাকম্যান, ব্লকবাস্টার ইত্যাদি ভিডিও গেম খেলেন, সেই এক একটি ভিডিও গেম প্রোগ্রাম লিখতে এক একটি প্রতিভাবান প্রোগ্রামারের কমপক্ষে তিনমাস সময় লাগে। আর সাধারণ কম্পিউটার ব্যবহারকারীর পক্ষে এ ধরনের প্রোগ্রাম লেখা এক রকম অসম্ভব ব্যাপারই বলা চলে। এতো সময়, পরিশ্রম এবং বুদ্ধি খরচের জন্য বর্তমানে কম্পিউটার যন্ত্রের থেকে সফট্‌ওয়্যারের দাম বহু গুণ বেশি।

বড় কাজের প্রোগ্রাম লেখা খুব কঠিন বলে সারা পৃথিবীতে প্রায় সমস্ত কম্পিউটার ব্যবহারকারীই বড় প্রোগ্রাম নিজে লেখার চেষ্টা করেন না। এই কাজে নিযুক্ত আছেন কয়েকটি প্রতিষ্ঠান, যাঁরা এক একটি সর্বজনোপযোগী প্রোগ্রামের সফট্‌ওয়্যার তৈরী করে সেই সফট্‌ওয়্যার ফ্লপি ডিস্কে কিম্বা অন্য ধরনের বাড়তি স্মৃতিতে রেকর্ড করে বাণিজ্যিক ভাবে বিক্রি করেন। এ ধরনের অ্যাপ্লিকেশন সফট্‌ওয়্যারকে বলা হয় সফট্‌ওয়্যার প্যাকেজ (Software Package) কিম্বা ক্যানড্‌ প্রোগ্রাম (Canned Program)। উপমাটি মন্দ নয়, ক্যানড্‌ ফ্রুট জুসের মতোই ক্যানড্‌ প্রোগ্রাম। ক্রেতাকে কোনোই চিন্তা করতে হয় না, শুধু চোখ বুজে কম্পিউটারের ফ্লপি ড্রাইভে কিম্বা ক্যাসেট রেকর্ডারে প্যাকেজটি ঢুকিয়ে চালিয়ে দিলেই কম্পিউটার কাজ শুরু করে দেয়।

এই ভাবে বিভিন্ন প্যাকেজ লাগিয়ে নিমেষের মধ্যে আমরা আমাদের কম্পিউটারকে একটি বুদ্ধিমান টাইপরাইটার, ভিডিও গেম খেলার মেশিন, টেলিভিশন কমার্শিয়ালে ব্যবহৃত বিচিত্র এফেক্ট সৃষ্টি করার যন্ত্র, এক্সরে স্ক্যান থেকে মস্তিষ্কের টিউমার হয়েছে কিনা তা নির্ধারণ করার ব্যবস্থা, স্কুলের বাচ্ছাদের পড়া শেখানোর মাস্টারমশাই, সুশিক্ষিত এবং সৎ অ্যাকাউনটেন্ট, তুখোড় ড্রাফটস্‌ম্যান, সঙ্গীতের

সুরকার, আধুনিক কবিতা লেখার কবি, জ্যোতিষী, কুয়াশাচ্ছন্ন রাত্রে প্লেন নিরাপদে অবতরণ করানোর পাইলট……আর তালিকা বাড়াবো না—এককথায় পৃথিবীর যে কোনো পেশার কাজ কর্মে পারদর্শী যন্ত্রে পরিণত করতে পারি।



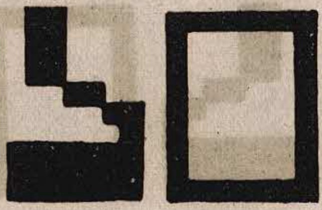
ইউটিলিটি সফট্ওয়্যার

এবার নিশ্চয়ই বুঝতে পারছেন, কম্পিউটার ব্যবহারকারীর কাছে অ্যাপ্লিকেশন সফট্‌ওয়্যারের গুরুত্ব ঠিক কতখানি? সত্যি কথা বলতে কি বর্তমানে প্রচলিত বিভিন্ন প্যাকেজগুলির সঙ্গে পরিচয় না থাকলে চিন্তাই করা যায় না একটি কম্পিউটার দিয়ে কত বিচিত্র রকম কাজ করানো সম্ভব।

ইউটিলিটি সফট্‌ওয়্যার

অ্যাপ্লিকেশন সফট্‌ওয়্যার লেখার সময় আমাদের প্রায়ই কয়েকটি বিশেষ রুটিন (Routine) ব্যবহার করতে হয়, যেমন একাধিক অজানা সংখ্যা নিয়ে কাজ করার সময় প্রায়ই সংখ্যাগুলিকে সারি এবং স্তম্ভে ভাগ করে লিখতে হয়। একে অঙ্কের ভাষায় বলা হয় ম্যাট্রিক্স (Matrix)। সেই ম্যাট্রিক্সের ইনভার্স (Inverse) কিম্বা ট্রান্সপোজ (Transpose) কিছু কিছু ক্ষেত্রে নির্ণয় করার প্রয়োজন হয়। এ ধরণের রুটিন যেহেতু সর্বদা সমস্ত প্রোগ্রামে কাজে লাগেনা, তাই কম্পিউটারের নির্মাতারা OS সফট্ওয়্যার তৈরী করার সময় এই সব রুটিন তাতে যুক্ত করেন না। অথচ OS এ যদি এমন আরো রুটিন সংযুক্ত থাকতো তবে অ্যাপ্লিকেশন সফট্‌ওয়্যার লেখা আরো সংক্ষিপ্ত ও সুবিধাজনক হয়ে যেত।

ঠিক এই কথা ভেবেই ইউটিলিটি (Utility) সফট্‌ওয়্যারের প্রচলন করা হয়েছে। ইউটিলিটি সফট্‌ওয়্যার হলো এই ধরণেরই সমস্ত রুটিন কিম্বা গণিতের ফর্মুলা সমাধানের প্রোগ্রাম, যেটি প্রোগ্রামার নিজে তৈরী করে নিতে পারেন অথবা কোনো সফট্‌ওয়্যার নির্মাতার কাছ থেকে ক্যানড্ প্রোগ্রাম হিসেবে কিনে নিতে পারেন। ইউটিলিটি প্রোগ্রাম গুলি সর্বদা বাড়তি স্মৃতি ব্যবস্থায় (যেমন ম্যাগনেটিক ক্যাসেট কিম্বা ফ্লপি ডিস্ক ইত্যাদিতে) লিপিবদ্ধ করে রাখা হয়। অ্যাপ্লিকেশন



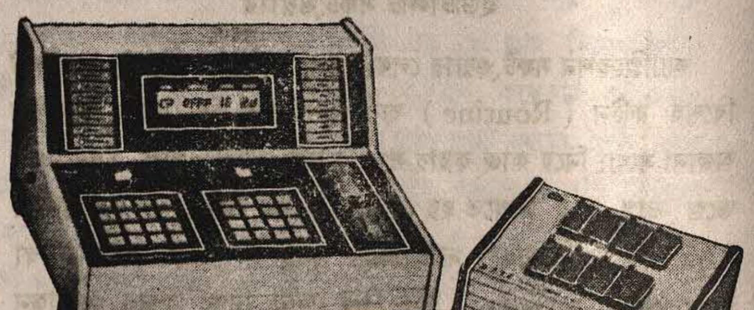
অ্যাপ্লিকেশন প্রোগ্রাম

চিত্র ৬৩

প্রোগ্রাম লেখার সময় যখনই রুটিনগুলির প্রয়োজন হয়, রুটিনটি না লিখে প্রোগ্রামার তার পরিবর্তে C.P.U.-কে নির্দেশ দেন বাড়তি স্মৃতি (Secondary storage) থেকে ওই রুটিনের প্রোগ্রাম পড়ে নিয়ে সেটি মূল অ্যাপ্লিকেশন প্রোগ্রামে সংযুক্ত (Merge) করে নিতে। C.P.U. তখন OS-এর সাহায্য নিয়ে বাড়তি স্মৃতি থেকে ইউটিলিটি প্রোগ্রামের রুটিনকে মূল প্রোগ্রামে ঢুকিয়ে নেয়। ফলে মূল প্রোগ্রামের সঠিক জায়গায় রুটিনের সফট্‌ওয়্যার লিখিত হয়ে যায় আর প্রোগ্রামারেরও কাজ হয়ে যায় অনেক সহজ ও সংক্ষিপ্ত।

শুধু মাত্র ম্যাট্রিক্সই নয় অন্যান্য বিভিন্ন কাজের উপযোগী সফট্‌ওয়্যার ইউটিলিটি প্রোগ্রাম হিসেবে সেকেণ্ডারি মেমরিতে লিখে রাখা হয়।

PROM Programmer যন্ত্র। সফট্‌ওয়্যার নির্মাতারা তাঁদের অ্যাপ্লিকেশন সফট্‌ওয়্যার **PROM**-এ রেকর্ড করতে এ ধরণের যন্ত্র ব্যবহার করেন। যন্ত্রটির সাহায্যে আটটি কিম্বা আরো বেশী সংখ্যক **PROM**-কে এক সঙ্গে প্রোগ্রাম করা যায়।



পরিচ্ছেদ ১১

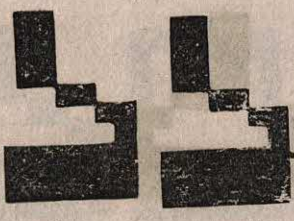
কম্পিউটার প্রোগ্রামিং করার ভাষা

লো লেভেল ল্যাংগোয়েজ

কম্পিউটার প্রোগ্রাম করার মূল ভাষা হলো লো লেভেল ল্যাংগোয়েজ (Low level language) বা মেশিন ল্যাংগোয়েজ (Machine language)। এই ভাষায় বাইনারি সংখ্যা ব্যবহার করে কম্পিউটার প্রোগ্রামিং করতে হয়। মেশিন ল্যাংগোয়েজের দুটি বিশেষ সুবিধা আছে। প্রথমটি হলো, এর সাহায্যে অত্যন্ত সংক্ষিপ্ত আকারে প্রোগ্রাম লেখা যায়, আর দ্বিতীয়টি হলো এই ভাষা C. P. U. সরাসরি বুঝতে পারে বলে মেশিন ল্যাংগোয়েজে লেখা নির্দেশ পালন করতে কম্পিউটার সব থেকে কম সময় নেয়।

কিন্তু মেশিন ল্যাংগোয়েজের যাবতীয় নির্দেশ, তথ্য সবই বাইনারি কিম্বা হেক্স সংখ্যার মাধ্যমে লেখা হয়। তাই প্রোগ্রামে কি নির্দেশ দেওয়া হয়েছে তা প্রোগ্রামে চোখ বোলালেই বোঝা যায় না। মেশিন ল্যাংগোয়েজ প্রোগ্রাম থেকে আসল নির্দেশকে খুঁজে বের করতে বিশেষ ইন্সট্রাকশন সেট (Instruction set) তালিকার সাহায্য নিতে হয়। প্রোগ্রাম লেখার সময়ও প্রোগ্রামারকে প্রতিনিয়ত এই ইন্সট্রাকশান সেটের শরণাপন্ন হতে হয়। ফলে এই ভাষায় প্রোগ্রামিং করা অপেক্ষাকৃত জটিল হয়ে পড়ে।

লো লেভেল ভাষার আর একটি বড় অসুবিধা হলো, বিভিন্ন



হাই লেভেল ল্যাংগোয়েজ

নির্মাতার তৈরী C. P. U.-এর আভ্যন্তরীণ সার্কিট আলাদা হওয়ায় তাদের ইন্সট্রাকশন সেট্ ও সম্পূর্ণ আলাদা হয়। ফলে কোনো প্রোগ্রামার যে কম্পিউটার ব্যবহার করতে অভ্যস্থ, তাঁকে যদি সেই কম্পিউটারের বদলে অন্য কম্পিউটার ব্যবহার করতে দেওয়া হয় (যাতে অন্য মডেলের C. P. U ব্যবহার করা হয়েছে), তাহলে তাঁকে আবার নতুন করে সেই C. P. U-টির ইন্সট্রাকশন সেটের সঙ্গে পরিচিত হতে হয়।



চিত্র ৬৪

C. P. U. Emulator যন্ত্র। কোনো **C. P. U.**-কে তার নিজস্ব ইন্সট্রাকশন সেট ছাড়া অন্য **C.P.U.**-র ইন্সট্রাকশন সেটের সাহায্যে প্রোগ্রাম করতে এই যন্ত্রটি ব্যবহার করা হয়। **Emulator** যন্ত্রটি **C. P. U.** এবং প্রোগ্রামারের মাঝে দোভাষীর মত কাজ করে। ফলে প্রোগ্রামার তাঁর পরিচিত ইন্সট্রাকশন সেটের সাহায্যে অপরিচিত ইন্সট্রাকশন সেট যুক্ত **C.P.U** প্রোগ্রাম করতে পারেন।

সুতরাং বুঝতেই পারছেন মেশিন ল্যাংগোয়েজ চট্ করে লেখা বেশ কঠিন কাজ। সাধারণ কম্পিউটার ব্যবহারকারী, যাঁদের যান্ত্রিক ব্যাপার-স্যাপার বোঝার বিষয়ে বিশেষ তাগিদ নেই, তাঁদের জন্য এই লো লেভেল ল্যাংগোয়োজ আদৌ উপযুক্ত নয়।

হাই লেভেল ল্যাংগোয়েজ

কম্পিউটারকে সর্বজন ব্যবহারোপযোগী করে তুলতে বেশ কয়েকটি সহজ আন্তর্জাতিক প্রোগ্রামিং ভাষা তৈরী করা হয়েছে। এই ভাষাতে বাইনারি সংখ্যার পরিবর্তে আমাদেরই বোধ্য বাক্য, হরফ এবং সংখ্যা ব্যবহার করে প্রোগ্রামটি লেখা হয়। তাছাড়া এই প্রোগ্রাম অপরিবর্তিত ভাবে (কোনো কোনো ক্ষেত্রে সামান্য রদবদল করে) বিভিন্ন নির্মাতার তৈরী কম্পিউটারে ব্যবহার করা যায়। পুরো প্রোগ্রামটি আমাদের বোধ্য ভাষায় (যেমন ইংরিজি, রাশিয়ান, জাপানী, ফরাসী ইত্যাদি) লেখা হয় বলে কোনো ইন্সট্রাকশন সেটের সাহায্য না নিয়েই একজন প্রোগ্রামার প্রোগ্রামটি লিখতে পারেন এবং অন্যের



ইন্টারপ্রিটার কম্পাইলার

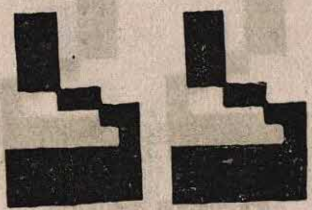
লেখা প্রোগ্রাম পড়ে সঙ্গে সঙ্গে তার অর্থ বুঝতে পারেন। এই ধরনের সমস্ত প্রোগ্রামিং ভাষাকে বলা হয় হাই লেভেল ল্যাংগোয়েজ (High level language)।

বিভিন্ন কাজের উপযোগী করে নানা রকম হাই লেভেল প্রোগ্রামিং ভাষা তৈরী করা হয়েছে। যেমন বৈজ্ঞানিক কাজের জন্য FORTRAN (FORmula TRANslation), ALGOL (ALGOrithmetic Language), বাণিজ্যিক কাজের জন্য COBOL (COmmon Buisness Oriented Language), সাধারণ সমস্ত কাজের জন্য BASIC (Beginner's All-purpose Symbolic Instruction Code), PL/1, PL/M, APL (A Programming Language), বিখ্যাত ফরাসী গণিতজ্ঞ পাসকাল-এর নামানুসারে PASCAL, FORTH, বিশেষ গ্রাফিক এফেক্ট সৃষ্টি করার জন্য C, কম্পিউটার নিয়ন্ত্রিত যন্ত্রপাতি চালাতে APT, বাচ্চাদের জন্য সহজ ভাষা LOGO, ডেটা বেস এবং ফোটো টাইপ সেটিং কাজের উপযোগী SNOBOL (StriNg Oriented SymBOlic Languge), পঞ্চম প্রজন্মের কম্পিউটারের জন্য তৈরী PROLOG, LISP প্রভৃতি।

এই সমস্ত ভাষার মধ্যে BASIC হলো সব থেকে জনপ্রিয়, এবং সমস্ত মাইক্রোকম্পিউটারে ব্যবহৃত আন্তর্জাতিক প্রোগ্রামিং ভাষা। আমরা কম্পিউটার প্রোগ্রামিং করার হাই লেভেল ভাষা হিসেবে BASIC-ই শিখবো। BASIC ভালোভাবে রপ্ত হলে, অন্যান্য আরো উন্নত এবং অপেক্ষাকৃত কঠিন কম্পিউটার ভাষা শেখা সহজ হয়ে যাবে, কারণ বেশীর ভাগ হাই লেভেল ভাষার মধ্যেই নিয়মগত অনেক সাদৃশ্য আছে।

ইন্টারপ্রিটার এবং কম্পাইলার

হাই লেভেল প্রোগ্রামিং ভাষায় আমাদের বোধ্য হরফ, বাক্য প্রভৃতি ব্যবহার করা হয়। এতে আমাদের কাজের খুব সুবিধে হয় বটে, কিন্তু কম্পিউটারের C. P. U এই প্রোগ্রাম বুঝতে পারেনা। C. P. U একমাত্র ইন্সট্রাকশন সেট্ অনুসারে লো লেভেল ভাষায় লেখা ডিজিটাল সংকেতই বুঝতে পারে। তাই সমস্ত কম্পিউটারেই



ইন্টারপ্রিটার কম্পাইলার

হাইলেভেল ভাষায় লেখা প্রোগ্রামের নির্দেশ C. P. Uতে পাঠাবার আগে লো লেভেল ভাষার প্রোগ্রামে অনুবাদ করে নেওয়া হয়। কম্পিউটারের OS-ই এই বিশেষ কাজটি করে থাকে।

অনুবাদ করতে OS-এ সাধারণত দুটি বিশেষ ধরনের সফট্‌ওয়্যার ব্যবহার করা হয়। প্রথমটিকে বলা হয় ইন্টারপ্রিন্টার (Interpreter) এবং দ্বিতীয়টি হল কম্পাইলার (Compliler)।

ইন্টারপ্রিটার সফট্‌ওয়্যারটি কম্পাইলার সফট্‌ওয়্যারের থেকে ছোট। এটিকে লিপিবদ্ধ করতে খুব বড় ROM স্মৃতি ভাণ্ডারের প্রয়োজন হয় না। ছোটো মাইক্রো কম্পিউটারে এবং পকেট কম্পিউটারে তাই ইন্টারপ্রিটার সফট্‌ওয়্যার বেশী ব্যবহার করা হয়। ইন্টারপ্রিটার যুক্ত কম্পিউটারে হাই লেভেলে লেখা অ্যাপ্লিকেশন প্রোগ্রাম চালানো (RUN) হলে C. P. U. অ্যাপ্লিকেশন প্রোগ্রামের প্রতি লাইনের নির্দেশ লো লেভেল বা মেশিন ল্যাংগোয়েজে অনুবাদ করে সঙ্গে সঙ্গে তা পালন করে। প্রতিটি লাইনের নির্দেশ অনুবাদ (Translate) ও পালন (Execute) করার পরই পরবর্তী লাইনের নির্দেশ অনুবাদ ও পালন করা হয় (অনেকটা পকেট ক্যালকুলেটরের মতো, বোতাম টিপলেই সঙ্গে সঙ্গে কাজ হয়)।

অপর পক্ষে কম্পাইলার যুক্ত কম্পিউটার সম্পূর্ণ অ্যাপ্লিকেশন প্রোগ্রামটিকে এক সঙ্গে হাই থেকে লো লেভেল ল্যাংগোয়েজে অনুবাদ করে নেয়।

তবে কম্পিউটারে কম্পাইলার বা ইন্টারপ্রিটার যাই লাগানো থাক্ না কেন, তাতে প্রোগ্রামিং পদ্ধতির কোনো পার্থক্য হয় না। কম্পাইলার লাগানো কম্পিউটারে ঠিক যে ভাবে প্রোগ্রাম করতে হয়, ইন্টারপ্রিটার লাগানো কম্পিউটারকেও ঠিক একই ভাবে প্রোগ্রাম করা হয়।

পরিচ্ছেদ ১২

BASIC

BASIC শব্দটি এসেছে বিগিনারশ্ অল পারপাস্ সিমবলিক ইন্সট্রাক্শন কোড্ (Beginners All purpose Symbolic Instruction Code) কথাটিকে সংক্ষিপ্ত করে।

1964 সালে Dartmouth College-এ প্রথম BASIC ভাষার উৎপত্তি। তারপর ক্রমশ এর আরো অনেক উন্নতি সাধন করা হয়। বর্তমানে প্রায় সমস্ত মাইক্রো কম্পিউটার নির্মাতাই তাঁদের কম্পিউটারে মূল BASIC-এর আরো পরিবর্তিত উন্নত রূপ ব্যবহার করছেন। প্রত্যেক নির্মাতারই নিজস্ব পরিবর্তিত বেসিক ভাষা আছে। যেমন Apple কম্পিউটার নির্মাতারা তাঁদের Apple সিরিজের কম্পিউটার প্রোগ্রাম করার জন্য Apple soft BASIC ভাষা তৈরী করেছেন। Radio Shack তাঁদের কম্পিউটারের জন্য তৈরী করেছেন Radio Shack BASIC। ঠিক তেমনি B. B. C, Sinclair, Comodore, Bull, Olivetti ইত্যাদি প্রখ্যাত নির্মাতারা তাঁদের কম্পিউটারের জন্য তৈরী করে নিয়েছেন তাঁদের নিজস্ব বেসিক ভাষা। এ সমস্ত ভাষাকে পরিবর্ধিত বেসিক বা এনহ্যান্সড্ বেসিক (Enhanced BASIC) বলা হয়। মূল BASIC-এর সঙ্গে

১২
BASIC

অবশ্য Enhanced BASIC-এর প্রধান কীওয়ার্ডগুলির খুবই মিল আছে। তাই মূল BASIC ভাষা জানা থাকলে যে কোনো কম্পিউটারের ম্যানুয়াল পড়ে অনায়াসে সেই কম্পিউটার প্রোগ্রাম করা সম্ভব হয়। যেমন শুদ্ধ হিন্দি জানা থাকলে ভারতবর্ষের যে কোনো প্রদেশবাসীর হিন্দি শুনে তার অর্থ উদ্ধার করা যায়, সেই রকম আর কি। তবে 'একঠো পান দেও' ধরনের বিচিত্র হিন্দির জ্ঞান নিয়ে যেমন হায়দ্রাবাদ কিম্বা লক্ষ্ণৌ-এর হিন্দি এবং উর্দু ভাষীদের সঙ্গে কথোপোকথন চালানো অসুবিধাজনক, ঠিক তেমনি কেবলমাত্র কোনো একটি বিশেষ কম্পিউটারের Enhanced BASIC dialect শিখে অন্য কম্পিউটার প্রোগ্রাম করা বেশ কঠিন হয়ে পড়ে।

আমরা এই পরিচ্ছেদে মূল BASIC ভাষাই শিখবো। Enhanced BASIC-এর বিভিন্ন নির্দেশ, যেমন--LOAD, EDIT, COPY, DRAW ইত্যাদি এখানে আলোচনা করা হবে না কারণ জনপ্রিয় কম্পিউটার নির্মাতাদের নিজস্ব Enhanced BASIC-এ এতো ফারাক থাকে যে, কোনো একটি কম্পিউটারের Enhanced BASIC-এ লেখা প্রোগ্রাম অন্য কম্পিউটারে সরাসরি চালানোই যায় না। তাই ওই সমস্ত নির্দেশগুলি কম্পিউটার ব্যবহার করার সময় অপারেটিং ম্যানুয়াল পড়ে জেনে নিতে হবে।

BASIC ভাষা শেখার আগে আসুন কম্পিউটারকে কি করে চালাতে হয়, জেনে নিই। আমরা এখানে অত্যন্ত জনপ্রিয় দুটি কম্পিউটারকে উদাহরণ হিসেবে ব্যবহার করবো। প্রথমটি হল B.B.C. ACORN কম্পিউটার। B.B.C. কম্পিউটার বর্তমানে ভারতের বেশীর ভাগ স্কুল ও কলেজে ব্যবহার করা হচ্ছে। আর দ্বিতীয়টি হলো Spectrum + কম্পিউটার। এই কম্পিউটারটি বোধ হয় পৃথিবীর সব থেকে জনপ্রিয় পারশোনাল কম্পিউটার। তবে অন্যান্য কোম্পানির কম্পিউটারের ক্ষেত্রেও পুরো ব্যাপারটি কিন্তু মোটামুটি একই রকম হবে।

ধরা যাক আমারা হাতের কাছে একটি B. B. C ACORN অথবা আমাদের দেশে তৈরী SCL UNICORN কম্পিউটার পেয়ে গেছি। দুটি কম্পিউটারই হুবহু একই রকম দেখতে এবং একই ভাবে কাজ করে। সেই সঙ্গে রয়েছে আর একটি বহুল ব্যবহৃত

কম্পিউটার Sinclair Spectrum+। কম্পিউটার ছাড়া আমাদের কাছে আছে আরো দুটি জিনিস, সে দুটি হল রঙিন টেলিভিশন সেট।

লক্ষ্য করে দেখুন কম্পিউটার থেকে টেলিভিশন অ্যানটেনার সকেটে লাগাবার একটা তারও আমাদেরও কাছে রয়েছে। এই তারটি TV-র অ্যানটেনার সকেটে গুঁজে দিয়ে টেলিভিশনের সুইচ অন্ করুন। তারপর B.B.C ACORN এবং Spectrum+এর পাওয়ার সাপ্লাই প্লাগ 220 ভোল্ট AC মেইন্‌সে সংযুক্ত করুন।

প্লাগ গোঁজা মাত্র B. B. C ACORN-এর স্পিকার থেকে বিপ্ করে একটা শব্দ শুনতে পাবেন। Spectrum-এর বেলায় কিন্তু কোনো শব্দ হবে না। সেই সঙ্গে দেখবেন B. B. C-র সঙ্গে সংযুক্ত টিভি স্ক্রিনের ওপরের বাঁ দিকে ফুটে উঠেছে Power on Message, "BBC Computer 16 K BASIC" আর Spectrum+এর সঙ্গে সংযুক্ত টিভি স্ক্রিনের নিচে ফুটে উঠেছে "(c) 1982 Sinclair Research Ltd"

যদি ছবি না ফোটে তবে রঙিন টিভির চ্যানেল পাল্টান। UHF এর 36 নম্বর চ্যানেলে ছবি আসবে। ছবি স্পষ্ট করতে ফাইন টিউন ঘোরান। যেহেতু টেলিভিশন সেটের স্পিকার থেকে বিশ্রী আওয়াজ হবে, তাই ভলিউম শূন্য করে দিন।

এবার একটা কিছু করা দরকার। Spectrum+এর কী বোর্ডে চোখ বোলান, দেখবেন কী বোর্ডে অনেক পরিচিত ইংরিজি শব্দ লেখা আছে, যেমন PRINT, FOR, NEXT, GOTO, READ ইত্যাদি। বেশ তাহলে Spectrum+এর কী বোর্ডের PRINT লেখা বোতামটাই টিপুন। অমনি দেখবেন স্ক্রিনের নিচে যে লেখাটি ফুটে উঠেছিল সেটি অদৃশ্য হয়ে তার জায়গায় PRINT শব্দটি ফুটে উঠেছে।

এবার B. B. C কী বোর্ড দেখুন। কি কাণ্ড! এখানে তো PRINT, FOR ইত্যাদি লেখা কোনো বোতামই নেই, তার বদলে শুধু A, B, C, D ইত্যাদি লেখা বোতাম আছে। কি আর করা যাবে P, R, I, N, T এই পাঁচটি বোতাম পর পর টাইপ করুন, দেখবেন স্ক্রিনের ওপরের বাঁদিকের লেখা মুছে গিয়ে সেখানে PRINT লেখাটি ফুটে উঠলো।

এবার Spectum + এবং B. B. C-তে 5, + এবং 4 লেখা বোতাম টিপুন। দেখবেন স্ক্রিনে "PRINT 5+4" লেখাটি ফুটে উঠেছে। B. B. C. র ক্ষেত্রে এই লেখাটি থাকবে স্ক্রিনের উপরে আর Spectrum + এর ক্ষেত্রে সেটি থাকবে স্ক্রিনের নিচে। এই ছোট্ট নির্দেশ বা Commandটি পড়ে বলুনতো আমরা কম্পিউটারকে কি করতে আদেশ দিচ্ছি? এতো জলের মতো সোজা, 5 এবং 4 কে যোগ করে যোগফলটি ছাপিয়ে দেওয়ার আদেশ। ব্যাস, এখানেই আমাদের প্রোগ্রাম শেষ।

এবার আমাদের কর্তব্য হলো কম্পিউটারকে জানিয়ে দেওয়া যে প্রোগ্রাম করা শেষ হয়েছে। এই কাজের জন্য B. B. C-র কী বোর্ডে Return লেখা বোতামটি টিপুন, আর Spectrum + এর কী বোর্ডে Enter লেখা বোতামটি টিপুন। ওই বোতাম টেপার সঙ্গে সঙ্গে কম্পিউটারের C. P. U. বুঝে যাবে প্রোগ্রাম করা শেষ হয়েছে।

B.B.C. কম্পিউটারের C. P. U. তখন তার অভ্যন্তরীণ ROM এ রক্ষিত কম্পাইলারের সাহায্যে এই হাই লেভেল প্রোগ্রামকে মেশিন ল্যাংগোয়েজে অনুবাদ করে নেবে আর Spectrum + এর C. P. U. তার স্মৃতি ভাণ্ডারের ROM এ রক্ষিত ইন্টারপ্রিটারের সাহায্যে এই অনুবাদের কাজটি করে ফেলবে। অনুবাদ হয়ে গেলে দুটি কম্পিউটারের C. P. U. আমাদের দেওয়া নির্দেশ অনুযায়ী ALU-তে 5 এবং 4 যোগ করে, যোগফল Data Bus মারফৎ ভিডিও ক্যারেকটার জেনারেটর পেরিফেরাল্‌সে পাঠাবে। এই পেরিফেরালস্‌টি C. P. U.-এর নির্দেশ অনুসারে টিভি স্ক্রিনে (5 এবং 4 এর যোগফল) 9 ফুটিয়ে তুলবে। কাজগুলি এতোই দ্রুত হবে যে বোতাম টেপার সঙ্গে সঙ্গে আমরা টিভি স্ক্রিনের ওপরে বাঁ দিকে 9 সংখ্যাটি ফুটে উঠতে দেখবো।

এবারে তাহলে BASIC ভাষা ব্যবহার করে কি ভাবে বড় বড় সব সমস্যা সমাধান করতে হয়, তার কায়দা শেখা যাক।

BASIC প্রোগ্রামের গঠন

যে কোনো হাই লেভেল প্রোগ্রাম লেখার সময় কয়েকটি বিশেষ নিয়ম মেনে চলতে হয়। BASIC ও এর ব্যাতিক্রম নয়। এই

নিয়ম না মানলে কম্পিউটার প্রোগ্রামটির কোনো নির্দেশই গ্রহণ করতে পারে না। BASIC ভাষা শেখার শুরুতে আমাদের তাই ভালো করে BASIC প্রোগ্রামের নিয়ম বা Syntax (উচ্চারণ সিনট্যাক্স) এবং গঠন অর্থাৎ structure-কে শিখে নিতে হবে। নিচে সঠিক ভাবে BASIC প্রোগ্রাম লেখার নিয়মগুলি দেওয়া হলো।

(১) প্রোগ্রামে লিখিত BASIC ভাষার প্রতিটা নির্দেশ, যাকে কম্পিউটার পরিভাষায় বলা হয় statement (স্টেটমেন্ট) অথবা Key word (কী ওয়ার্ড), তাদের আলাদা আলাদা লাইনে লিখতে হবে।

(২) প্রতিটি স্টেটমেন্ট একটি লাইনেই সম্পূর্ণ হতে হবে। (অবশ্য এই লাইনটি যত খুশি বড় হতে পারে, এবং তার জন্য যদি সেটি VDU স্ক্রিনে একাধিক লাইন গ্রহণ করে, তবে সেটি কিন্তু নিয়ম বিরুদ্ধ হবে না। কম্পিউটার ওই একাধিক লাইনকে একই লাইন হিসেবে গণ্য করবে)

(৩) প্রতিটি লাইন শুরু করার আগে একটি লাইন নাম্বার (Line number) লিখতে হবে। এই লাইন নাম্বারকে statement number (স্টেটমেন্ট নাম্বার) ও বলা হয়। স্টেটমেন্ট নাম্বার সর্বদাই ধণাত্মক পূর্ণ সংখ্যা হবে। BASIC প্রোগ্রামে 1 থেকে 99999 পর্যন্ত স্টেটমেন্ট নাম্বার দেওয়া যেতে পারে। এর থেকে ছোটো কিম্বা বড় কোনো সংখ্যা স্টেটমেন্ট নাম্বার হিসেবে লিখলে কম্পিউটার সেই লাইন বুঝতে পারবে না।

(৪) প্রতি লাইনে আলাদা স্টেটমেন্ট নাম্বার দিতে হবে। একই স্টেটমেন্ট নাম্বার দিয়ে একাধিক স্টেটমেন্ট লাইন লিখলে কম্পিউটার ঠিক মতো কাজ করবে না।

(৫) একমাত্র Command (কম্যাণ্ড) বা আদেশ কী ওয়ার্ড ব্যবহার করার সময় স্টেটমেন্ট নাম্বার দিতে হবে না।

(৬) সব থেকে ছোট স্টেটমেন্ট নাম্বার লিখিত লাইনটি কম্পিউটার সর্বাগ্রে পড়বে এবং তাতে দেওয়া নির্দেশ পালন করবে। নির্দেশ পালন পূর্ণ হলে, কম্পিউটার ঠিক পরের বড় স্টেটমেন্ট নাম্বার যুক্ত লাইনটির নির্দেশ পালন করবে। এই ভাবে ক্রমে ছোট থেকে বড় স্টেটমেন্ট নাম্বার যুক্ত লাইনগুলির নির্দেশ কম্পিউটার পালন করে যাবে।

১২

BASIC

(৭) ছোট থেকে বড় স্টেটমেন্ট নাম্বার লেখার সময়ে, দুটি পর পর (Consequtive) স্টেটমেন্ট নাম্বারের মধ্যে যেমন খুশি ব্যবধান রাখা যাবে। যেমন—যদি কোনো প্রোগ্রামে মোট চারটি লাইন থাকে তবে সেই চারটি লাইনের স্টেটমেন্ট নাম্বার আমরা 1, 2, 3, 4 লিখতে পারি আবার 17, 20, 550, 999 ও লিখতে পারি, শুধু লক্ষ্য রাখতে হবে চারটি নাম্বার যেন ক্রমশ ছোট থেকে বড় ভাবে সাজানো হয়।

সাধারণত BASIC প্রোগ্রাম লেখার সময় সমস্ত স্টেটমেন্ট নাম্বারের মধ্যে 10 করে ব্যবধান রাখা হয়। এই ব্যবধান রাখার ফলে, পরে দরকার পড়লে যে কোনো দুটি স্টেটমেন্ট লাইনের ফাঁকে আমরা 9টি বাড়তি স্টেটমেন্ট লাইন ঢোকাতে পারি।

(৮) স্টেটমেন্ট নাম্বারের পরই স্টেটমেন্ট বা কী ওয়ার্ড লিখতে হবে।

(৯) কী ওয়ার্ডের পর লিখতে হবে কী ওয়ার্ডের আনুসঙ্গিক তথ্য। তবে সমস্ত স্টেটমেন্টের সঙ্গেই যে তথ্য থাকতে হবে এমন কোনো বাধ্যবাধকতা নেই।

(১০) কী ওয়ার্ড সর্বদা ক্যাপিটাল (Capital letter) হরফে টাইপ করতে হবে।

(১১) BASIC প্রোগ্রামে স্টেটমেন্ট নাম্বার, কী ওয়ার্ড এবং তথ্যের মাঝে যেমন খুশি ফাঁক রাখা রাখা যেতে পারে, এমন কি ইচ্ছে করলে আদৌ কোনো ফাঁক না রাখলেও চলে। তবে প্রোগ্রামটি যাতে সহজে পড়া যায়, তার জন্য স্টেটমেন্ট নাম্বার, কী ওয়ার্ড এবং তথ্যের মাঝে এক হরফ করে ফাঁকা জায়গা রাখতে হবে।

BASIC প্রোগ্রাম লেখার সময়ে এই এগারোটি নিয়ম সর্বদা মনে রাখতে হবে। বইতে দেওয়া অজস্র BASIC প্রোগ্রামের উদাহরণ গুলি দেখলেই পুরো ব্যাপারটি ক্রমশ আপনার কাছে পরিষ্কার হয়ে যাবে। আপাতত এই নিয়মগুলি এক সঙ্গে মনে রাখতে একটি মডেল প্রোগ্রামের লাইন মুখস্ত করে রাখুন।

```
99999 PRINT "Basic"
```

ওপরের লাইনে 99999 হলো স্টেটমেন্ট নাম্বার। এখানে BASIC ভাষার সর্বোচ্চ স্টেটমেন্ট নাম্বারটি লেখা হয়েছে যাতে

একই সঙ্গে সর্বোচ্চ স্টেটমেন্ট নাম্বারটিও আপনার মনে থাকে (নিয়ম ৩)।

স্টেটমেন্ট নাম্বারের পর এক হরফ পরিমান জায়গা ফাঁক (space) রাখা হয়েছে (নিয়ম ১১)।

স্টেটমেন্ট নাম্বারের পরই লেখা হয়েছে একটি BASIC কী ওয়ার্ড PRINT (নিয়ম ৮)।

কী ওয়ার্ডটি সম্পূর্ণই ক্যাপিট্যাল হরফে লেখা হয়েছে (নিয়ম ১০)।

কী ওয়ার্ডের পরে আবার এক হরফ জায়গা ফাঁকা রাখা. হয়েছে (নিয়ম ১১)।

ফাঁকা জায়গার পর লেখা হয়েছে "Basic"। এটি হলো আনুসঙ্গিক এবং ঐচ্ছিক (Optional) তথ্য (নিয়ম ৯)।



কি ভাবে প্রোগ্রামিং করা শিখবো

BASIC প্রোগ্রামের গঠন (Structure) আমরা জেনে গেছি। এবার চলুন কম্পিউটারে কী বোর্ডের সামনে বসে হাতে নাতে প্রোগ্রামিং করা শিখে নিই। টাইপ করার অভিজ্ঞতা থাকেলে বড় জোর ঘন্টা দুই লাগবে BASIC ভাষা শিখতে। তারপর চর্চা করে বুদ্ধিতে ধার দেওয়ার দায়িত্ব কিছু আপনার!

কম্পিউটার টিভি কিম্বা V.D.U.-র সঙ্গে লাগিয়ে অন্ করুন। কম্পিউটারের মডেল নিয়ে চিন্তা করবেন না, Apple II, Commo-

BASIC

dore 64, Sinclair QL, Spectrun + কিম্বা B. B. C Acorn যাই হোক না কেন, আমাদের প্রোগ্রাম সব কম্পিউটারেই চলবে। এমন কি Casio PB 100, PB 110, PB 200, Sharp কিম্বা Hewlett Packerd-এর পকেট কম্পিউটার হলেও কাজ চলবে।

পাওয়ার অন্ করলেই V. D. U, TV কিম্বা অন্যান্য ডিসপ্লেতে (যেমন পকেট কম্পিউটারের L. C. D. ডিসপ্লে) Power ON message ফুটে উঠবে। বিভিন্ন কোম্পানি অনুসারে এই message ও বিভিন্ন হবে। তবে এর সঙ্গে আমাদের প্রোগ্রামিং করার ব্যাপারে কোনো তফাৎ হবে না। আমরা একে একে সমস্ত BASIC কী ওয়ার্ডের সঙ্গে পরিচিত হবো, সেই সঙ্গে প্রতিটি কী ওয়ার্ড ব্যবহার করে এই বইয়ে দেওয়া অজস্র প্রোগ্রাম কম্পিউটারে চালিয়ে হাতে নাতে প্রোগ্রামিং শিখে নেব। প্রতিটি কী ওয়ার্ডের সঙ্গে ভাল মতো পরিচয় লাভ করার পর বইয়ে শেষে দেওয়া প্রশ্নের উত্তরটি কিন্তু অবশ্যই লিখে ফেলবেন। এইভাবে শিখলে দেখবেন BASIC ভাষা শিখতে এক দিনের বেশী সময় লাগবে না। তবে BASIC ভাষা ভাল করে রপ্ত করতে আপনাকে অবশ্যই রোজ কম্পিউটারে বসে চর্চা চালিয়ে যেতে হবে। BASIC ভালো করে শেখার আরো একটি চমৎকার উপায় হলো, যখনই কোনো BASIC প্রোগ্রাম দেখবেন তখুনি মনে মনে ভাববেন এই প্রোগ্রামটিকে যে ভাবে লেখা হয়েছে সে ভাবে না লিখে, আর কি অন্য কোনো ভাবে লেখা যায় না? এ ভাবে চিন্তা করতে করতে দেখবেন খুব অল্প দিনের মধ্যেই আপনি BASIC-এ দারুন ব্যুৎপত্তি তৈরী করে ফেলেছেন।

PRINT

কম্পিউটারের মাধ্যমে কোনো মন্তব্য, তথ্য কিম্বা উত্তর V.D.U-তে ফুটিয়ে তুলতে কিম্বা প্রিন্টারে ছাপিয়ে ফেলতে PRINT স্টেটমেন্ট ব্যবহার করা হয়। আমরা ইতি মধ্যেই PRINT কম্যাণ্ডের সঙ্গে পরিচিত হয়েছি, এবার দেখা যাক PRINT কম্যাণ্ড এবং স্টেটমেন্ট কেমন করে কাজে লাগাতে হয়।

১২

PRINT

কম্যাণ্ড হিসেবে PRINT ব্যবহার

BASIC ভাষায় কম্যাণ্ড হলো আদেশ এবং স্টেটমেন্ট হলো নির্দেশ। আদেশ আর নির্দেশের মধ্যে তফাৎ হল, আদেশ পেলেই কম্পিউটার সঙ্গে সঙ্গে অন্য সব কাজ কামিয়ে সেই আদেশ পালন করে। BASIC-এ স্টেটমেন্ট এবং কম্যাণ্ডের মধ্যে পার্থক্য বোঝাতে কম্যাণ্ড লাইনে স্টেটমেন্ট নাম্বার ব্যবহার করা হয় না। দ্বিতীয় তফাৎ হলো, কম্পিউটার কম্যাণ্ডের আদেশ পালন করার পর সেই আদেশটিকে মনে রাখে না। অপরপক্ষে যে কোনো স্টেটমেন্টে দেওয়া নির্দেশ পালন করার পরও কম্পিউটার স্মৃতিতে পুরো স্টেটমেন্টটি লিপি বদ্ধ হয়ে থাকে। সুতরাং কম্পিউটারকে ক্যালকুলেটরের মতো ব্যবহার করতে চাইলে কম্যাণ্ড ব্যবহার করতে হয়। PRINT কী ওয়ার্ডটি BASIC ভাষায় কম্যাণ্ড এবং স্টেটমেন্ট, দুভাবেই ব্যবহার করা চলে।

কম্যাণ্ডের উদাহরণ

আমাদের কম্পিউটার তো চালু আছে। এবার নিচে দেওয়া লাইনটি টাইপ করে স্ক্রিনে ফুটিয়ে তুলুন।

```
PRINT 4+2
```

তারপর আপনার কম্পিউটারের ENTER কিম্বা RETURN লেখা বোতামটি টিপুন। দেখবেন TVতে ফুটে উঠলো 4 এবং 2-এর যোগফল 6। একই ভাবে নিচের কম্যাণ্ড গুলি টাইপ করে যান। প্রতি লাইন টাইপ করার পর ENTER কিম্বা RETURN বোতাম টিপতে কিন্তু ভুলবেন না। এটা না টিপলে কম্পিউটার হাই লেভেল কম্যাণ্ডকে লো লেভেলে অনুবাদ করবে না।

```
PRINT 4-2
```

এবার ENTER কিম্বা RETURN টিপুন, দেখবেন স্ক্রিনে উত্তর '2' ফুটে উঠবে।

এবার টাইপ করুন।

```
PRINT 4*2
```

ENTER কিম্বা RETURN টিপুন, স্ক্রিনে 4 এবং 2 গুণফল '8' ফুটে উঠবে।

৫২

PRINT
লাইব্রেরী ফাংশন

PRINT 4/2

এবার ENTER কিম্বা RETURN টিপুন, স্ক্রিনে 4÷2-এর ভাগফল '2' ফুটে উঠবে।

PRINT কম্যাণ্ডের সাহায্যে কম্পিউটারকে সায়েন্টিফিক ক্যালকুলেটরের মতো ব্যবহার।

আমরা এতোক্ষণ সহজ যোগ, বিয়োগ, গুণ এবং ভাগ করেছি। এবার সায়েন্টিফিক ক্যালকুলেটরে করা যায় এমন কঠিন অঙ্ক আমরা কম্পিউটারে করবো। এই অঙ্ক কষার আগে আমাদের BASIC ভাষার লাইব্রেরী ফাংশন (Library Function) গুলির সঙ্গে পরিচিত হতে হবে।

লাইব্রেরী ফাংশন

ফাংশন (Function) বলতে বোঝায় একটি গাণিতিক নিয়ম যার সাহায্যে কোনো একটি সংখ্যা (যাকে গাণিতিক ভাষায় বলে argument অথবা operand) থেকে আর একটি সংখ্যা বা ফলাফল (result) পাওয়া যায়। আমরা নিজেরাইতো কতো ফাংশন জানি, যেমন বর্গমূল (square root), বর্গফল (square power), ট্রিগোনোমেট্রির SIN, COS, TAN ইত্যাদি ইত্যাদি। এগুলি সবই বিশেষ অঙ্কের নিয়ম। কোনো সংখ্যার ওপর এই নিয়ম প্রয়োগ করলে একটি উত্তর পাওয়া যায় আর সেটিই হলো ফলাফল বা result.

সায়েন্টিফিক ক্যালকুলেটরে এ ধরণের বহু ফাংশনের বোতাম থাকে। ঠিক তেমনি BASIC ভাষাতেও, কম্পিউটারের OS-এ বহু গাণিতিক ফাংশনের ফর্মুলা ভরা থাকে। এই পূর্ব পূর্বনির্ধারিত ফাংশন গুলিকে বলা হয় লাইব্রেরী ফাংশন। কয়েকটি কম্পিউটার কী বোর্ডে এই লাইব্রেরী ফাংশনের জন্য আলাদা আলাদা বোতাম থাকে, যেটি টিপলেই V.D.U. তে লাইব্রেরী ফাংশনের সংকেত ফুটে উঠে, আর অন্যান্য কম্পিউটারে ইংরিজি হরফের বোতাম টিপে লাইব্রেরী ফাংশনের সংকেত টাইপ করতে হয়। নিচে BASIC ভাষায় যে কটি লাইব্রেরী ফাংশন পাওয়া যায় তার একটি সারণী দেওয়া হলো।

লাইব্রেরী ফাংশন	প্রয়োগ	গাণিতিক অর্থ
ABS (ABSolute value	ABS (−4·6)	$\lvert -4{\cdot}6 \rvert = 4{\cdot}6$
SGN (SiGNum)	SGN (A)	যদি A=O হয় তবে SGN A=O যদি A+ve সংখ্যা হয় তবে SGN A= + যদি A−ve সংখ্যা হয় তবে SGN A= −
SQR (SQuare Root)	SQR(2)	$\sqrt{2}$
SIN (SINe)	SIN (2)	Sin 2
COS (CoSine)	COS (2)	Cos 2
TAN (TANgent)	TAN (2)	Tan 2
ASN (Arc SiNe)	ASN (1)	arc Sin 1
ACS (Arc Co Sine)	ACS (1)	arc Cos 1
ATN (Arc TaN)	ATN (1)	are Tan 1
LN (Lograrithm Natural)	LN (2)	$\text{Log}_e 2$
EXP (EXPonen-tial)	EXP (2)	e^2
INT (INTeger)	INT (4·6)	Integer part of 4·9=5 (দশমিক চিহ্ন তুলে দিয়ে সব থেকে কাছাকাছি মানের পূর্ণ সংখ্যায় প্রকাশ করা)

১২

PRINT

লাইব্রেরী ফাংশন	প্রয়োগ	অর্থ
PI	3 * PI	$2 \times \pi =$ $2 \times 3{\cdot}1415927$
RND (RaNDom)	RND (O)	যা খুশি উল্টো পাল্টা RANDOM সংখ্যা তৈরী করার নিদেশ।

যোগ, বিয়োগ, গুণ, ভাগ এবং লাইব্রেরী ফাংশন ছাড়া আরও একটি গাণিতিক ক্রিয়ার ব্যবস্থা BASIC-এ আছে, সেটি হলো '↑', এই চিহ্নের অর্থ হলো—

$$2 \uparrow 3 = 2^3 = 2 \times 2 \times 2$$

অনেক কম্পিউটারে আবার '↑'-এর পরিবর্তে '∧' কিম্বা '**' চিহ্ন ব্যবহার করা হয়।

এবার এই লাইব্রেরী ফাংশন ব্যবহার করে কয়েকটি অঙ্ক কষে ফেলা যাক।

অঙ্ক : $\left\{\frac{6{\cdot}36 + (4{\cdot}4)^{6{\cdot}3}}{3{\cdot}7}\right\} \times \log_e 4 = ?$

কী বোর্ডে টাইপ করুন

PRINT (6·36+4·4∧6·3)/3·7*LN(4)

ENTER কিম্বা RETURN টিপুন, দেখবেন উত্তর হবে

4242·7708

অঙ্ক : $\frac{3^{12} + e^{10}}{\pi} = ?$

কী বোর্ডে টাইপ করুন—

PRINT (3∧12+EXP(10))/PI

ENTER কিম্বা RETURN টিপুন, V.D.Uতে উত্তর পাবেন

176174·174

PRINT কম্যাণ্ড ব্যবহার করে কম্পিউটারকে টাইপ রাইটারের মতো ব্যবহার।

শুধু অঙ্কই নয়, PRINT কম্যাণ্ড ব্যবহার করে আমরা যে কোনো কম্পিউটার ক্যারেকটারকে V.D.U স্ক্রিনে ফুটিয়ে তুলতে পারি।

52 PRINT

আর সঙ্গে যদি প্রিন্টার কিম্বা ইলেকট্রনিক টাইপ রাইটার লাগানো থাকে, তবে প্রিন্টারে কিম্বা টাইপরাইটারে ওই ক্যারেকটারগুলি ছাপা হয়ে যায়।

PRINT কম্যাণ্ড লেখার পর যে ক্যারেকটার কিম্বা বাক্যটি V.D.Uতে ফুটিয়ে তুলতে হবে বা প্রিন্টারে ছাপাতে হবে, সেটিকে কোটেশন চিহ্নের " " মধ্যে লিখতে হয়। এই চিহ্নের অন্তরগত যে কোন বিষয়বস্তু কম্পিউটার হুবহু V.D.U স্ক্রিনে ফুটিয়ে তোলে কিম্বা প্রিন্টারে ছাপিয়ে দেয়।

কী বোর্ডে টাইপ করুন

PRINT "Hi!!"

এবার RETURN কিম্বা ENTER টিপুন, অমনি দেখবেন স্ক্রিনে কিম্বা প্রিন্টারের কাগজে ফুটে উঠেছে।

Hi!!

আবার PRINT লিখে " " চিহ্নের মধ্যে যা মন চায় তাই টাইপ করে যান। তারপর ENTER কিম্বা RETURN টিপুন, দেখুন কি হয়।

PRINT স্টেটমেন্টের ব্যবহার

একটু আগে আমরা PRINT কম্যাণ্ড নিয়ে পরীক্ষা নিরিক্ষা করলাম। দেখলাম PRINT কম্যাণ্ড ব্যবহার করে কম্পিউটারকে এক লাইনের বেশী নির্দেশ দেওয়া যায় না। তাই ওভাবে কোনো বড় প্রোগ্রাম লেখা সম্ভব নয়। বড় প্রোগ্রাম লিখতে গেলে আমাদের ব্যবহার করতে হয় PRINT স্টেটমেন্ট। স্টেটমেন্ট লেখার নিয়ম আমরা BASIC প্রোগ্রামের গঠন থেকেই জেনেছি। এবার হাতে নাতে পরীক্ষা করে দেখা যাক :—

টাইপ করুন—

10 PRINT "A=2"

তারপর ENTER কিম্বা RETURN টিপুন। এবার কিন্তু আর আগের মতো A=2 কথাটি V.D.Uতে আলাদা করে ফুটে উঠবে না, বরং পুরো স্টেটমেন্টটাই V.D.U স্ক্রিনে আগে যেমন ছিলো তেমনই থাকবে। Spectrum কম্পিউটারে ENTER বোতাম টেপার পর 10 PRINT "A=2" কথাটি স্ক্রিনের নিচ থেকে

PRINT

একদম স্ক্রিনের উপরে উঠে যাবে, আর B.B.C ACORN-এর ক্ষেত্রে Cursorটি (V.D.U স্ক্রিনে ক্রমাগত জ্বলতে নিভতে থাকা ছোট্ট আলোর বিন্দুটি) 10 PRINT "A=2" লাইন থেকে সরে ঠিক নিচের লাইনের এক দম বাঁ দিকে স্থান নেবে।

আবার টাইপ করুন

```
20 PRINT "B=3"
```

ENTER কিম্বা RETURN টিপুন। (এবার থেকে কিন্তু প্রতিবার এক একটি লাইন টাইপ করার পর 'ENTER কিম্বা RETURN টিপুন' কথাটি আর বলবো না। ওটা আপনাকে মনে করে করে যেতে হবে)।

নিচের লাইনগুলো পরপর টাইপ করে যান—

```
30  PRINT "A+B="; 2+3
40  PRINT
50  PRINT "A-B="; 2-3
60  PRINT
70  PRINT "A*B="; 2*3
80  PRINT
90  PRINT "A/B="; 2/3
100 END
```

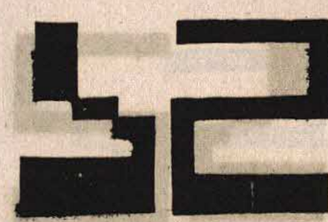
স্টেটমেন্ট 100 অবধি টাইপ করার পর টাইপ করুন RUN

এটা টাইপের পর যেই ENTER কিম্বা RETURN টিপবেন, অমনি দেখবেন V.D.Uতে ফুটে উঠেছে।

```
A=2
B=3
A+B=5
A-B=-1
A*B=6
A/B=·66666666
```

তাহলে সম্পূর্ণ প্রোগ্রাম এবং উত্তরটি হলো—

```
 10  PRINT "A=2"
 20  PRINT "B=3"
 30  PRINT "A+B="; 2+3
 40  PRINT
 50  PRINT "A—B="; 2—3
 60  PRINT
 70  PRINT "A*B="; 2*3
 80  PRINT
 90  PRINT "A/B="; 2/3
100  END
RUN
A=2
B=3
A+B=5
A—B=—1
A*B=6
A/B=·66666666
```



PRINT

প্রোগ্রামটি থেকে পরিষ্কার দেখা যাচ্ছে ৬নং নিয়ম অনুসারে কম্পিউটার ছোট থেকে বড় স্টেটমেন্ট নাম্বারের নির্দেশ পালন করে গেছে। প্রথমেই লাইন নাম্বার 10-এর " " চিহ্নের মধ্যে লেখা "A=2"কে স্ক্রিনে ফুটিয়েছে তুলেছে। তারপর লাইন নাম্বার 20তে দেওয়া নির্দেশ অনুযায়ী 'B=3' কথাটি লিখেছে। একইভাবে লাইন নাম্বার 30তে দেওয়া নির্দেশ অনুসারে 'A+B=' ছাপার পর 2+3-এর যোগফল ফুটিয়েছে। লাইন নাম্বার 40তে PRINT স্টেটমেন্টের পরে কিছুই নেই। তাই V.D.U স্ক্রিনে ওই লাইনটি ফাঁকা রেখেছে।

তারপর আবার লাইন নাম্বার 50তে দেওয়া নির্দেশ মতো কাজ করে গেছে।

স্টেটমেন্ট নাম্বারের মধ্যে ফাঁক রাখার সুবিধে

আমাদের প্রোগ্রাম কিন্তু একটু ত্রুটি থেকে গেছে। কম্পিউটার যে ছয় লাইন উত্তর ফুটিয়ে তুলেছে তার শেষ চার লাইনের মাঝে

52

PRINT

এক লাইন করে ফাঁক রাখা হয়েছে। কিন্তু প্রথম তিনটি লাইনের মধ্যে কোনো ফাঁক রাখা হয়নি। এই ত্রুটি খুব সহজেই ঠিক করা যায়, এর জন্য কী বোর্ডে টাইপ করুন।

```
15 PRINT
25 PRINT
LIST
```

LIST টাইপ করার পর ENTER কিম্বা RETURN টিপুন, দেখবেন আমাদের আগে টাইপ করা প্রোগ্রামটিই পুনরায় স্ক্রিনে ফুটে উঠেছে, আর তার 10, 20 এবং 20, 30 স্টেটমেন্ট লাইনের মাঝে 15 PRINT এবং 25 PRINT লাইন দুটি ঢুকে গেছে।

```
10 PRINT "A=2"
15 PRINT
20 PRINT "B=3"
25 PRINT
30 PRINT "A+B="; 2+3
40 PRINT
50 PRINT "A-B="; 2-3
60 PRINT
70 PRINT "A*B="; 2*3
80 PRINT
90 PRINT "A/B="; 2/3
100 END
```

এরপর টাইপ করুন

```
RUN
```

তারপর ENTER কিম্বা RETURN টিপুন। V.D.Uতে এবার দেখা যাবে—

```
A=2
B=3
A+B=5
A—B=—1
A*B=6
A/B='66666666
```

52
PRINT

তাহলেই দেখুন স্টেটমেন্ট নাম্বারগুলির মধ্যে 10 নম্বরের ব্যবধান রাখার জন্য কত সহজে একটি করে বাড়তি স্টেটমেন্ট 10, 20 এবং 20, 30 লাইনের মাঝে ঢুকিয়ে দেওয়া গেল। আমরা যদি স্টেটমেন্ট নাম্বারগুলি 10, 20, 30 ইত্যাদির পরিবর্তে 1, 2, 3, দিতাম, তবে কিন্তু এভাবে সংশোধন করার কোনো সম্ভাবনাই থাকতো না। সে ক্ষেত্রে আবার সম্পূর্ণ প্রোগ্রামটি নতুন করে টাইপ করতে হতো।

PRINT স্টেটমেন্টে কমা এবং সেমিকোলোন ব্যবহার

আর একটি প্রোগ্রাম টাইপ করুন

```
NEW
10 PRINT "Demonstration program of the
          function of Comma and Semicolon in
          PRINT Statement"
   20 PRINT
   30 PRINT
   40 PRINT "A" ; "B" ; "C" ; "D"
   50 PRINT
   60 PRINT "A" ; ; "B" ; ; "C" ; ; "D"
   70 PRINT
   80 PRINT "A", "B", "C", "D"
   90 END
   RUN
```

```
Demonstration program of the function of
Comma and Semicolon in PRINT Statement.

   ABCD
   A B C D
   A         B         C         D
```

PRINT স্টেটমেন্টের মধ্যে যে বিষয়বস্তুগুলি কম্পিউটার ছাপবে বা V.D.Uতে ফুটিয়ে তুলবে, সেগুলির মাঝে যদি ' ; ' ব্যবহার করা

১২

RUN, LIST, NEW

হয় তবে ছাপার বিষয়বস্তুগুলির মাঝে কোনো ফাঁক থাকবেনা (যেমন ABCD)। একাধিক ‘ ; ’ চিহ্ন পর পর বসিয়ে অবশ্য ফাঁক সৃষ্টি করা যায়। তবে বড় ফাঁক সৃষ্টি করতে গেলে ‘ , ’ চিহ্ন ব্যবহার করতে হয়।

মনে রাখবেন, “ ” চিহ্নের ভিতরে লেখা ‘ , ’ কিম্বা ‘ ; ’ চিহ্নের কিন্তু কোনো ফাঁক নিয়ন্ত্রণের ক্ষমতা থাকেনা। “ ” চিহ্নের অন্তর্গত বিষয়বস্তু টাইপ করার সময়ে SPACE বোতাম টিপে ক্যারেকটার-গুলি মাঝে যেমন খুশি ফাঁক রাখা যেতে পারে (যেমন সাধারণ টাইপ রাইটারে করা হয়)। ছাপার সময় কিম্বা V.D.Uতে ফুটিয়ে তোলার সময় কম্পিউটার “ ” চিহ্নের অন্তর্গত পুরো বিষয়বস্তুটি হুবহু ফুটিয়ে তুলবে।

RUN, LIST, NEW

আগের প্রোগ্রামে আমরা তিনটি নতুন বেসিক কম্যাণ্ডের সঙ্গে পরিচিত হয়েছি। সে তিনটি হলো RUN, LIST এবং NEW কম্পিউটারে সম্পূর্ণ BASIC প্রোগ্রাম টাইপ করা পর কম্পিউটারকে ওই প্রোগ্রাম অনুসারে কাজ শুরু করতে বলার আদেশ বা কম্যাণ্ড হলো RUN. কী বোর্ডে RUN টাইপ করে ENTER কিম্বা RETURN বোতাম টিপলেই কম্পিউটারের C.P.U, ইন্টারপ্রিন্টার কিম্বা কম্পাইলারের সাহায্যে হাই লেভেল প্রোগ্রামকে লো লেভেলে অনুবাদ করে নিয়ে নির্দেশ পালন করতে শুরু করে।

LIST এই কম্যাণ্ডটি হলো কম্পিউটারের RAM-এ সঞ্চিত যাবতীয় তথ্যকে V.D.U স্ক্রিনে ফুটিয়ে তোলার আদেশ। ধরুন আপনি কম্পিউটারে একটি প্রোগ্রাম Load করেছেন এবং সেই প্রোগ্রাম অনুসারে কম্পিউটারকে RUN করিয়েছেন। ফলে বর্তমানে স্ক্রিন থেকে আপনার প্রোগ্রামটি মুছে গিয়ে তার জায়গায় কম্পিউটারের উত্তরটি দেখা যাচ্ছে। এই অবস্থায় আপনি যদি আবার মূল প্রোগ্রামটি ফিরে দেখতে চান তবে LIST কথাটি টাইপ করে ENTER কিম্বা RETERN বোতাম টিপুন, অমনি সঙ্গে সঙ্গে স্ক্রিনে পরপর সঠিকক্রমে সাজানো প্রোগ্রামটি ফুটে উঠবে।

New ক্যালকুলেটরের যেমন All Clear (AC) বোতাম থাকে, BASIC ভাষায় NEW কম্যাণ্ডের কাজও ঠিক তাই। কম্পিউটার কী বোর্ডে NEW কথাটি টাইপ করে ENTER কিম্বা RETURN টিপলেই কম্পিউটারের RAM স্মৃতিতে রক্ষিত পূর্বের সমস্ত প্রোগ্রাম মুছে যায়। কম্পিউটারে যখনই কোনো নতুন প্রোগ্রাম টাইপ করতে হয়, তখন পূর্বের প্রোগ্রামের সঙ্গে যাতে নতুন প্রোগ্রাম না মিশে যায়, তার জন্য সর্বদা নতুন প্রোগ্রামের শুরুতে NEW কথাটি টাইপ করে নিতে হয়।

১২

INPUT

INPUT

এখন প্রশ্ন উঠতে পারে, কম্পিউটারে অঙ্ক কষতে গেলে যদি প্রতিবারই আগা গোড়া পুরো ফর্মুলা টাইপ করতে হয় তবে সাধারণ প্রিন্টার লাগানো ক্যালকুলেটরের থেকে কম্পিউটারে কি আর এমন সুবিধে পাওয়া যায়?

ঠিক এই কথা ভেবেই BASIC ভাষায় INPUT স্টেটমেন্টের ব্যবস্থা করা হয়েছে। INPUT স্টেটমেন্টের সাহায্যে আমরা কম্পিউটারে Numaric (নিউমেরিক) এবং String (স্ট্রিং) Variable (ভেরিয়েবল)-এর নাম ঢোকাতে পারি। Numeric এবং String বলতে কি বোঝায়, সে বিষয়ে পরে আসছি। তার আগে দেখা যাক INPUT স্টেটমেন্ট ব্যবহারের সুবিধা কতো।

ছোটবেলায় কষা পাটিগণিতের সেই সব অঙ্ক মনে আছে তো? সেই যে পাঁচজন মজুর একটা কাজ দশ দিনে করতে পারে। তাহলে পনেরো জন মজুর সেই কাজ কত দিনে করবে? এই সব অঙ্ক ছোট বেলায় কি দুর্বোধ্যই না লাগতো। পাটিগণিত বইয়ের প্রশ্নমালায় একই অঙ্ককে ঘুরিয়ে-ফিরিয়ে মজুরের বদলে পুরুষ, স্ত্রী, বালক ইত্যাদি বসিয়ে এবং পাঁচ, দশ, পনেরোর বদলে অন্য সংখ্যা বসিয়ে দেওয়া থাকতো তবু কষতে গিয়ে মাঝে মধ্যে ভুল হোত। ইস্, সেই সময় যদি

কখনো যদি কম্পিউটারের বিদ্যুৎ সরবরাহ মুহূর্তের জন্যও বিচ্ছিন্ন হয়, তাহলেও কিন্তু NEW কম্যাণ্ডেরই কাজ হয়। অর্থাৎ এতো খেটে-খুটে যা কিছু প্রোগ্রাম করা হয়েছে, মুহূর্তের মধ্যেই সব মুছে RAM স্মৃতি ভাণ্ডার ফাঁকা হয়ে যায়। তাই কম্পিউটার ব্যবহার করার সময় সার্ভো স্টেবিলাইজার ব্যবহার করা উচিত। তাছাড়া কম্পিউটারের প্লাগ যাতে মেইনস সকেটে ঠিক মতো লাগে (লগবগ না করে), সেদিকে খেয়াল রাখা উচিত।

52
INPUT

একটা কম্পিউটার পাওয়া যেতো, তবে INPUT স্টেটমেন্ট ব্যবহার করে অঙ্কের মূল ফর্মুলা কম্পিউটারে ঢুকিয়ে দিতাম আর বিভিন্ন অঙ্ক অনুসারে পাঁচ, দশ, পনেরো ইত্যাদিগুলি কী বোর্ডে টাইপ করে ঝটপট পাঁচ মিনিটের মধ্যে পুরো প্রশ্নমালার অঙ্ক কষে উত্তর ছাপিয়ে সাংঘাতিক কাণ্ড করে ফেলতাম! তাহলে দেখা যাক ওই সব অঙ্ক কিভাবে আমরা INPUT স্টেটমেন্টের সাহায্যে করতে পারি।

এ ধরণের অঙ্কে সাধারণত তিনটি জিনিসের মান বলে দেওয়া থাকে যার সাহায্যে একটি অজানা মান বার করতে হয়। যেমন—

1.	5 জন মজুর	একটা কাজ করতে পারে।
2.	10 দিনে	তবে সেই কাজ কত দিনে
3.	15 জন মজুর	করতে পারবে ?

অঙ্কটি খাতায় কষতে হলে আমরা লিখবো—

5 জন লোক একটি কাজ 10 দিনে করে

1টি লোক সেই কাজ 10×5 „ „

15 জন লোক সেই কাজ $\frac{10 \times 5}{15}$ দিনে করে

সাধারণ PRINT কম্যাণ্ড দিয়ে আমরা এই অঙ্ক করবো এই ভাবে—

```
PRINT (10*5)/15
```

কিন্তু প্রশ্নমালার অন্য একটি অঙ্ক করতে গেলে আমাদের আবার খাতায় অঙ্কটির কিছুটা কষে তারপর PRINT কম্যাণ্ড দিয়ে উত্তর পেতে হবে। এই অসুবিধার হাত থেকে রেহাই পেতে আমরা এবার INPUT স্টেটমেন্ট ব্যবহার করবো।

```
NEW
10 INPUT A
20 INPUT B
30 INPUT C
40 PRINT (A*B)/C
50 END
```

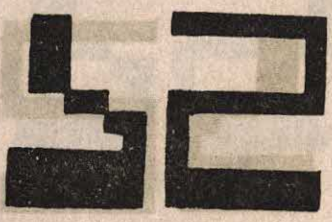
এই প্রোগ্রামে 5 জন মজুরের স্থানে A, 10 দিনের স্থানে B এবং 15 জন মজুরের স্থানে C ধরা হয়েছে।

প্রোগ্রামটি RUN করলে আমরা দেখতে পাবো V.D.U স্ক্রিন পরিষ্কার হয়ে গেল আর স্ক্রিনে একটি '?' চিহ্ন ফুটে উঠলো। এর অর্থ, কম্পিউটার আমাদের কাছে A-র মান জানতে চাইছে। কী বোর্ডে টাইপ করুন '5', তারপর ENTER কিম্বা RETURN টিপুন। দেখবেন 5 উধাও হয়ে আবার '?' চিহ্ন ফুটে উঠেছে। অর্থাৎ এবার কম্পিউটার B-র মান জানতে চাইছে। সুতরাং কী বোর্ডে B-র মান 10 টাইপ করে ENTER বা RETURN টিপুন। ঠিক আগের মতই কম্পিউটার এবার C-র মান জানতে চাইবে। তাই কী বোর্ডে C-র মান 15 টাইপ করে ENTER কিম্বা RETURN টিপুন। এতক্ষণে কম্পিউটার A, B, C সবকটি Variable-এর মানই জেনে গেছে। তাই এবার স্ক্রিনে '?' চিহ্নের বদলে $\frac{5\times10}{15}$-এর উত্তর ফুটে উঠবে।

পুরো প্রোগ্রামটি এবং কম্পিউটারের ছাপানো উত্তর সম্পূর্ণই নিচে দেওয়া হলো—

```
  U     NEW
  U     10 INPUT A
  U     20 INPUT B
  U     30 INPUT C
  U     40 PRINT (A*B)/C
  U     50 END
  U     RUN
C/U     ?5
C/U     ?10
C/U     ?15
  C     3·3333333
```

কি খুব সহজ না? আসুন, এবার প্রশ্নমালার আর একটি অঙ্ক কষা যাক। ধরা যাক অঙ্কটি হলো—



INPUT

একটি পূর্ণাঙ্গ প্রোগ্রামের কোন্ অংশটি প্রোগ্রামার টাইপ করছেন এবং কোন্ অংশটি কম্পিউটার ছেপে দিচ্ছে কিম্বা T. V. স্ক্রিনে ফুটিয়ে তুলছে সেটি পাঠকের বোঝবার সুবিধার জন্য এবার থেকে পরপর কয়েকটি প্রোগ্রামের একেবারে বাঁদিকে আমরা দুটি হরফ ব্যবহার করবো। এ দুটি হল U এবং C. যে লাইনের পাশে U আছে, সেটি প্রোগ্রামারকে টাইপ করতে হবে আর যেটির পাশে C আছে, সেটি কম্পিউটার স্বয়ংক্রিয় ভাবে ছেপে দেবে। C/U লেখা লাইনের বাঁদিকের হরফটি ছাপবে কম্পিউটার এবং ডানদিকের হরফ বা বাক্যটি টাইপ করবেন প্রোগ্রামার। যেমন C/U ? 5 লাইনের '?' চিহ্নটি কম্পিউটার ফুটিয়ে তুলবে এবং 5 সংখ্যাটি প্রোগ্রামার টাইপ করবেন। মনে রাখবেন, কম্পিউটার প্রোগ্রামের সঙ্গে কিন্তু এই সাংকেতিক হরফের কোনোই সম্পর্ক নেই।

১২

নিউমেরিক/স্ট্রিং ভেরিয়েবল্

7 জন বালক একটি কাজ 20 দিনে করে
14 জন বালক সেই কাজ কত দিনে করবে ?

এই অঙ্কে A = 7, B = 20 এবং C = 14

আমরা এবার কম্পিউটারে টাইপ করবো RUN (মূল প্রোগ্রাম আমরা কম্পিউটারে আগেই ঢুকিয়ে দিয়েছি, তাই শুধু RUN টিপলেই কম্পিউটার আগের প্রোগ্রাম মতো কাজ শুরু করবে)।

U	RUN
C/U	? 7
C/U	? 20
C/U	? 14
C	10

আরো একটা অঙ্ক কষার ইচ্ছে থাকলে আবার RUN টিপে A, B, Cর তথ্য টাইপ করুন।

নিউমেরিক এবং স্ট্রিং ভেরিয়েবল্

এই ছোট্ট প্রোগ্রামে A, B, Cকে বলা হয় Input numeric Variable. এই Variable-এর মান কেবল গাণিতিক সংখ্যা হতে পারে যেমন—

A = 1, 2, −65·3 ইত্যাদি।

(আপাতত জেনে রাখুন A থেকে Z অবধি 26টি হরফকে নিউমেরিক ভেরিয়েবল হিসেবে ব্যবহার করা যায়।)

এই ভেরিয়েবল ছাড়া BASIC প্রোগ্রামে আর এক ধরনের ভেরিয়েবল ব্যবহার করা হয়, যাকে বলা হয় String Variable। String Variable-এর কিন্তু কোনো গাণিতিক মান হয় না। এক মাত্র যে কোনো ইংরিজি হরফ, বাক্য এবং যোগ, বিয়োগ ইত্যাদি স্পেশাল ক্যারেকটারই String Variable-এর মান হতে পারে। ব্যাপারটি আরো পরিষ্কার হবে যদি আমরা String Variable ব্যবহার করে একটা প্রোগ্রাম RUN করি।

String Variable লিখতে হলে নিউমেরিক ভেরিয়েবলের পরে একটা ডলার চিহ্ন '$' বসাতে হয়, যেমন A হলো নিউমেরিক ভেরিয়েবল আর A $ হলো স্ট্রিং ভেরিয়েবল।

উদাহরণ

ঐকিক নিয়মের অঙ্ক কষার প্রোগ্রামটি এখনো আমাদের কম্পিউটারে ভরা আছে, তবে V.D.U. স্ক্রিনে এই মুহূর্তে প্রোগ্রামের বদলে কম্পিউটারের বলে দেওয়া উত্তরগুলি ফুটে আছে। তাই মূল প্রোগ্রামটি ফিরে দেখতে আমরা LIST কম্যাণ্ড ব্যবহার করবো।

```
U  LIST
C  10 INPUT A
C  20 INPUT B
C  30 INPUT C
C  40 PRINT ( A*B ) / C
C  50 END
```

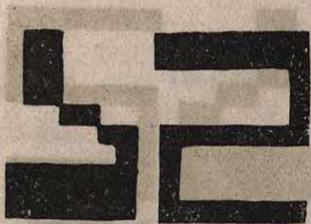
এবার নিচের লাইনগুলি টাইপ করুন

নোতুন টাইপ করা প্রোগ্রামগুলি 10, 20, 30 এবং 40 নম্বর স্টেটমেন্টের ফাঁকে নিজে থেকেই ঢুকে যাবে। ইচ্ছে হলে RUN টাইপ করার আগে LIST টিপে দেখে নিতে পারেন।

```
U   5 INPUT A$
U  15 INPUT B$
U  25 INPUT C$
U  31 PRINT A$ ; "=" ; A
U  32 PRINT B$ ; "=" ; B
U  33 PRINT C$ ; "=" ; C
U  34 PRINT
U  35 PRINT "therefore, the answer is"
U  36 PRINT
U  RUN
C  ?
```

এর অর্থ কম্পিউটার '?' চিহ্ন দিয়ে A$ এর মান জানতে চাইছে। A$ হলো মজুর বা বালকের নাম। সুতরাং পরপর টাইপ করে যান—

```
C/U  ? Workers
C/U  ? 7
```



User Friendly Software

C/U ? Days
C/U ? 20
C/U ? Workers
C/U ? 14

ব্যাস এর পর থেকে কম্পিউটারই স্ক্রিনে ফুটিয়ে তুলবে

C Workers =7
C Days =20
C Workers =14
C Therefore, the answer is
C 10.

এবার অঙ্কটা কিছুটা বোধগম্য হলো। প্রথমে 7 জন কাজের লোক 20 দিনে কাজটা করতো। পরে 14 জন কাজের লোক সেই কাজ করবে 10 দিনে।

কম্পিউটারকে দিয়ে প্রশ্ন করাতে হলে

এই প্রোগ্রামে একটা অসুবিধা কিন্তু এখনো রয়ে গেছে। সেটি হলো, কম্পিউটার যখন নিউমেরিক বা স্ট্রিং ভেরিয়েবল এর মান জানতে চাইছে, তখন বোঝাই যাচ্ছে না কখন কোন ভেরিয়েবল এর মান জানতে চাওয়া হচ্ছে।

তাই.এবার আমরা প্রোগ্রামটি আর একটু ঘুরিয়ে লিখবো।

```
U   1 PRINT  "Type the name of the first
              variable after the question mark"
U   6 PRINT  "Type the number of" ;; A$ ;;
              "after the question mark"
U  11 PRINT  "Type the name of the second
              variable after the question mark"
U  16 PRINT  "Type the number of" ;; B$ ;;
              "after the question mark"
U  21 PRINT  "Type the name of third variable
              after the question mark"
```

বর্তমানে প্রায় সমস্ত বাণিজ্যিক সফ্‌টওয়্যারই User Friendly করে তৈরী করা হয়। User Friendly সফ্‌টওয়্যারের বৈশিষ্ট্য হলো, এটি প্রোগ্রামারকে কোনো সমস্যা সমাধান করতে প্রতিপদে সাহায্য করে। এ-ধরনের সফ্‌টওয়্যার রচনার অজস্র উদাহরণ এই বইতে রয়েছে। পাটীগণিত অঙ্কের সাধারণ সফ্‌টওয়্যারটি কিভাবে ধাপে ধাপে একটি User Friendly সফ্‌টওয়্যারে রূপান্তরিত করা হয়েছে তার পরিচয় এ পাতার সফ্‌টওয়্যারটি থেকেই পাবেন।

```
U 26 PRINT "Type the number of" ;; C$ ;; "after question mark"
U 31
U 32
U 33
```

এতো দূর টাইপ করার পর পুরো প্রোগ্রামটা কি দাঁড়ালো দেখা যাক—

```
U LIST
C 1 PRINT "Type the name of the first variable after the question mark"
C 5 INPUT A$
C 6 PRINT "Type the numbers of" ;; A$ ;; "after the question mark"
C 10 INPUT A
C 11 PRINT "Type the name of the second variable after the question mark"
C 15 INPUT B$
C 16 PRINT "Type the number of" ;; B$ ;; "after the question mark"
C 20 INPUT B
C 21 PRINT "Type the name of the third variable after the question mark"
C 25 INPUT C$
C 26 PRINT "Type the number of" ;; C$ ;; "after the question mark"
C 30 INPUT C
```

```
C  34  PRINT
C  35  PRINT  "Therefore, the answer is"
C  36  PRINT
C  40  PRINT  ( A*B )/C
C  50  END
```

এবার প্রোগ্রামটি RUN করা যাক—

```
   U   RUN
   C   Type the name of first variable after the
          question mark
C/U   ? Workers
   C   Type the number of workers after the
          question mark
C/U   ? 7
   C   Type the name of the second variable after
          the question mark.
C/U   ? Days
   C   Type the number of Days after the question
          mark
C/U   ? 20
   C   Type the name of the third variable after
          the question mark
C/U   ? Workers
   C   Type the number of Workers after the
          question mark
C/U   ? 14
   C   Therefore the answer is
   C   10
```

এবারে ব্যাপারটা আরো বোধগম্য হলো, তাই না ? শুধু তাই নয়, কম্পিউটারই আমাদের কাছে প্রশ্ন করে করে সব তথ্য জেনে নিল, ঠিক যেন একটি জীবন্ত যন্ত্র !!

১২

প্রোগ্রামের ভুল সংশোধন

এই হলো আমাদের প্রথম User friendly software-এর সঙ্গে পরিচয়। User friendly software লিখলে কম্পিউটার এমনই জীবন্ত মানুষের মতো আচরণ করে, যে, অনেক সময় যাঁরা আগে কখনো কম্পিউটার ব্যবহার করেননি, তাঁরা একেবারে চমৎকৃত হয়ে যান। আমরা এর পরে আরো ভালো ভালো User friendly প্রোগ্রাম চালিয়ে দেখবো। কিন্তু তার আগে এই প্রোগ্রামটিকে আর একবার ভাল করে দেখা যাক।

প্রোগ্রামে ভুল সংশোধনের নিয়ম

একটু পাতা উল্টে ১৬৩ পৃষ্ঠায় দেখুন,

আমরা পূর্বে যে প্রোগ্রাম করেছিলাম তাতে লাইন নাম্বার 31, 32 এবং 33-এ টাইপ করেছিলাম।

```
31  PRINT  A $ ; "=" ; A
32  PRINT  B $ ; "=" ; B
33  PRINT  C $ ; "=" ; C
```

কিন্তু পরে যখন প্রোগ্রামটিকে আরো উন্নত করলাম, তখন 31, 32, 33 লাইনে টাইপ করলাম (১৬৫ পৃষ্ঠা দেখুন)

```
31
32
33
```

পুরো প্রোগ্রাম টাইপ করার পর যখন LIST করলাম (১৬৫ পৃষ্ঠা) তখন দেখা গেল নতুন পরিবর্তিত প্রোগ্রামের 31, 32 এবং 33 লাইন সম্পূর্ণ উধাও হয়ে গেছে এবং সেই মতো প্রোগ্রাম RUN করার পর T.V. স্ক্রিনে '? 14' ফুটে ওঠার পর এক লাইন ফাঁকা রেখে কম্পিউটার দ্বিতীয় লাইনে 'Therefore···" ফুটিয়ে তুলেছে। এর থেকে বোঝা যাচ্ছে কম্পিউটারে পূর্বে যে প্রোগ্রাম ভরা ছিল, সেই প্রোগ্রামের একই লাইন নাম্বারে নতুন তথ্য টাইপ করতে কম্পিউটার পুরোনো তথ্য ভুলে গিয়ে নতুন তথ্যটি মনে রখেছে।

এই ব্যাপারটি খুবই গুরুত্বপূর্ণ। প্রোগ্রামে ভুল থাকলে বা সংশোধন করার কাজে সুবিধার জন্যই BASIC ভাষায় এ ব্যবস্থাটি রাখা হয়েছে।

52

END

প্রোগ্রামে যাই লেখা হোক না কেন, পরে ইচ্ছে করলে পূর্বে প্রোগ্রাম করা স্টেটমেন্ট নাম্বারের লাইন টাইপ করে তাতে অন্য স্টেটমেন্ট জুড়ে ENTER কিম্বা RETURN টিপলেই নতুন স্টেটমেন্টটি ওই লাইনে চলে যায়, আর পুরনো স্টেটমেন্টটি RAM স্মৃতি থেকে মুছে যায়।

এইভাবে ইচ্ছে করলে যে কোনো স্টেটমেন্ট লাইনই আগাগোড়া প্রোগ্রাম থেকে মুছে দেওয়া যেতে পারে। যেমন 34 এবং 36 লাইনে লেখা স্টেটমেন্টটি যদি একেবারেই প্রোগ্রাম থেকে সরিয়ে দিতে হয় তবে কী বোর্ডে টাইপ করুন—

```
34
36
```

BASIC ভাষায় স্টেটমেন্ট ছাড়া শুধু স্টেটমেন্ট নাম্বার গণ্য করা হয়না। তাই 34, 36 টাইপ করে ENTER বা RETURN টেপার সঙ্গে সঙ্গে মূল প্রোগ্রাম থেকে ওই দুটি লাইন বেমালুম অদৃশ্য হয়ে যায়। LIST কম্যাণ্ডের সাহায্যে আবার করে পুরো প্রোগ্রামের লিস্ট মিলিয়ে দেখে নিন সত্যি সত্যিই 34 এবং 36 নম্বর লাইন প্রোগ্রাম থেকে উধাও হয়েছে কিনা।

END

BASIC প্রোগ্রামে একেবারে শেষ স্টেটমেন্ট লাইনে END স্টেটমেন্ট লিখতে হয়। END স্টেটমেন্ট থেকে কম্পিউটার বুঝে যায় প্রোগ্রামের এইখানেই শেষ। অবশ্য ইদানীং বেশীর ভাগ আধুনিক কম্পিউটারেই END স্টেটমেন্টটি বাধ্যতামূলক নয়। যদি END নাও লেখা হয়, তাহলেও বহু কম্পিউটারে প্রোগ্রামটি ঠিকভাবে RUN করে। আমি যে সমস্ত পারশোনাল কম্পিউটার ব্যবহার করেছি (ছটি বিভিন্ন মডেল) তার কোনটাতেই END স্টেটমেন্ট বাধ্যতামূলক নয়। বস্তুত STOP স্টেটমেন্ট ব্যবহার করলে তবেই END স্টেটমেন্টের প্রয়োজন হয়। যাই হোক তবু মূল BASIC ভাষার নিয়মানুসারে আমাদের সমস্ত প্রোগ্রামের শেষে END স্টেটমেন্ট

দেওয়া হলো। তবে আপনার কম্পিউটারের dialect (কম্পিউটার নির্মাতার নিজস্ব BASIC)-এ যদি END স্টেটমেন্টের ব্যবস্থা একদমই না থাকে তবে শেষ স্টেটমেন্ট লাইনটি বাদ দিয়ে প্রোগ্রামটি টাইপ করবেন।

52

LET

LET

পকেট ক্যালকুলেটরে আমরা যে কোন সংখ্যা Memoryতে ধরে রাখতে পারি। যখনই আবার সংখ্যাটি দরকার হয়, Memory Recall বোতাম টিপলেই ডিসপ্লেতে সংখ্যাটি ফুটে ওঠে। অবশ্য ক্যালকুলেটর একটির বেশী সংখ্যা মনে রাখতে পারেনা। সে দিক দিয়ে কম্পিউটার কিন্তু দারুন কাজের। এটি শয়ে শয়ে সংখ্যা, এমন কি বাক্যও মনে রাখতে পারে। আর যখন যেটি দরকার, বোতাম টিপলেই V.D.U স্ক্রিনে হাজির হয়।

এই কাজ করার জন্য আমাদের নিউমেরিক বা স্ট্রিং ভেরিয়েবলের সাহায্য নিতে হয়। আমরা যদি ভেরিয়েবল্‌গুলির বিভিন্ন মান ঠিক করে, তাদের RAM স্মৃতিতে রেখে দিই, তবে যখন যেমন দরকার সেই ভেরিয়েবলকে PRINT করার নির্দেশ দিলেই আমাদের প্রয়োজনীয় তথ্য V.D.U স্ক্রিনে ফুটে ওঠে। এছাড়া, বহু প্রোগ্রামে আমাদের হামেসাই নিউমেরিক এবং স্ট্রিং ভেরিয়েবলের একটি নির্দিষ্ট মান ঠিক করে দেওয়ার প্রয়োজন হয়। এই কাজের জন্য BASIC ভাষায় LET স্টেটমেন্ট / কম্যাণ্ড ব্যবহার করা হয়। কম্পিউটার পরিভাষায় LET স্টেটমেন্টকে Assignment স্টেটমেন্ট বলে, কারণ এই স্টেটমেন্টের সাহায্যে, আমরা কোন ভেরিয়েবলের একটি নির্দিষ্ট মান ঠিক করে অর্থাৎ assign করে দিতে পারি।

আমরা এখন কম্পিউটার কি ভাবে বোতাম টিপে প্রোগ্রাম করতে হয় বুঝে গেছি। তাই এবার থেকে 'বোতাম টিপুন', 'ENTER কিম্বা RETURN টিপুন', 'LIST বা RUN করুন' ইত্যাদি ধরনের কথা আর লিখবো না। শুধু প্রোগ্রামটি এবং সেটি RUN করার পর কি দাঁড়ালো সেটিই এবার থেকে লেখা হবে। সেই

১২

LET

সঙ্গে প্রোগ্রামের কোন অংশটি আপনি টাইপ করছেন এবং কোন অংশটি কম্পিউটার TV স্ক্রিনে ফুটিয়ে তুলছে তাও আর U এবং C হরফের সংকেত ব্যবহার করে দেখিয়ে দেব না।

নিচে LET স্টেটমেন্টের বিভিন্ন ব্যবহার দেখানো হলো।

LET স্টেটমেন্টের সাহায্যে দুটি সংখ্যার যোগ, বিয়োগ, গুণ এবং ভাগফল বার করা—

```
NEW
10  LET  A=3
20  LET  B=6
30  PRINT  "A+B=" ; A+B
40  PRINT  "A-B=" ; A-B
50  PRINT  "A*B=" ; A*B
60  PRINT  "A/B=" ; A/B
70  END
RUN
A+B=9
A-B=-3
A*B=18
A/B=0·5
```

LET স্টেটমেন্ট বা কম্যাণ্ডটি ব্যবহার করা খুবই সহজ। আমরা অঙ্ক কষার সময় যেমন লিখি ধরা যাক **$x=2$** ঠিক তেমনি কম্পিউটারে x-এর মান 2 স্থির করতে গেলে লিখতে হয় **LET $X=2$.** এই নির্দেশ পেলে কম্পিউটার X নিউমেরিক ভেরিয়েবল্-এর মান 2 বলে ধরে নেয়।

LET স্টেটমেন্টের সাহায্যে ফারেনহাইট থেকে সেন্টিগ্রেট/সেলসিয়াস বার করার প্রোগ্রাম—

```
NEW
10 PRINT "Program to convert farenheit to degree celsius"
20 PRINT
30 PRINT "Type the temp in degree farenheit after question mark"
40 INPUT F
50 LET C=(5*(F—32) )/9
60 PRINT
```



LET

```
70 PRINT F ;; "degree farenheit=" ;; C ;;
"degree celsius"
80 END
RUN
Program to convert farenheit to degree celsius
Type the temp in degree farenheit after question
mark
? 32
32 degree farenheit=0 degree celsius
```

LET স্টেটমেন্টের সাহায্যে সেন্টিগ্রেট / সেলসিয়াস থেকে ফারেনহাইট বার করার প্রোগ্রাম ঠিক আগের মতোই হবে শুধু সেলসিয়াসের জায়গায় লিখতে হবে ফারেনহাইট। তাই এবার INPUT ভেরিয়েবল হবে C এবং ফর্মুলা হবে F=9*C/5+32 সুতরাং প্রোগ্রামটি হবে—

```
NEW
30
40 INPUT C
50 LET F=9*C/5+32
70 PRINT C ;; "degree celsius=" ;; F ;; "degree
farenheit"
```

LET স্টেটমেন্টের সাহায্যে বাঁদরের তৈলাক্ত বাঁশ বেয়ে ওঠার অঙ্ক।

প্রশ্ন : একটা বাঁদর তৈলাক্ত বাঁশ বেয়ে সেকেণ্ডে 2 ফুট ওপরে ওঠে আর সঙ্গে সঙ্গে 1·5 ফুট হড়কে যায়। তাহলে 5 ফুট লম্বা তৈলাক্ত বাঁশে চড়তে তার কত সময় লাগবে ?

এই প্রোগ্রাম অনুসারে পাটীগণিত বইয়ের সমস্ত তেলা বাঁশের অঙ্ক কষে ফেলা যাবে ! !

52

REM

```
10INPUT C,S,H
20 PRINT"C=RATE OF CLIMB=",C
30 PRINT"S=RATE OF SLIP=",S
40 PRINT"H=HEIGHT OF POLE=",H
50LET X=C-S
60LET T=H/X
70PRINT
80PRINT "TIME REQUIRED=",T,"SECOND
90END
>RUN
72,1.5,5
C=RATE OF CLIMB=                2
S=RATE OF SLIP=               1.5
H=HEIGHT OF POLE=               5

TIME REQUIRED=           10SECOND
```

চিত্র ৬৪

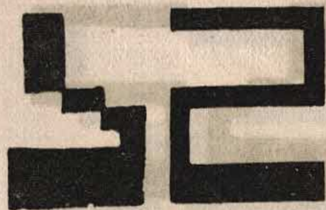
প্রোগ্রামটি B. B. C Microতে Run করার পর ইংক জেট্ প্রিন্টারের প্রিন্ট আউট।

REM

REMARK কথাটিকে সংক্ষিপ্ত করে REM কথাটির উৎপত্তি। প্রোগ্রামের অন্তর্গত বিভিন্ন স্টেটমেন্টের কার্যকলাপকে বিস্তারিতভাবে ব্যাখ্যা করার জন্য এই স্টেটমেন্ট ব্যবহার করা হয়। কম্পিউটারকে কোনো কাজ করার নির্দেশ REM স্টেটমেন্ট দেয় না।

REM স্টেটমেন্ট লেখার পর আমরা যা খুশি তাই সেই লাইনে টাইপ করে যেতে পারি। কম্পিউটার REM স্টেটমেন্ট যুক্ত লাইনকে প্রোগ্রাম লাইন হিসেবে গ্রাহ্য করেনা। REM বাদ দিয়ে আর যে সমস্ত স্টেটমেন্ট থাকে, কম্পিউটার সেই সব স্টেটমেন্টের নির্দেশ মতো কাজ করে।

তাহলে REM স্টেটমেন্টের সুবিধা কি? অনেক সময় বড় জটিল প্রোগ্রামের কোন্ স্টেটমেন্ট লাইন কি কাজ করছে, তা স্বয়ং প্রোগ্রামারেরই গুলিয়ে যায়। তাই প্রোগ্রামের মাঝে মাঝে REM স্টেটমেন্টের সাহায্যে মন্তব্য জুড়ে দিলে পুরো ব্যাপারটা আরো সহজে বোধগম্য হয়। পরবর্তী প্রোগ্রাম থেকে সহজেই বুঝতে পারবেন REM স্টেটমেন্ট কিভাবে ব্যবহার করা হয়।



REM

```
NEW
10 REM Demonstration program  to  show the
20 REM usage of REM Statement. REM Stands
for the word :—
30 REMARK.  So you can see, instead of typing
40 REM, You can also type REMARK.  Any
50 REM character or word can be typed after
REM, or you
60 REM can space out your remark as shown
70 REM below :—
80 REM
90 REM                    Hi !! How are you ?
100 REM                   *  *  *  *  *  *
110 REM
120 REM
130 REM You can even use empty REMs as you
140 REM have used empty PRINTs before.
150 REM We will now end this program.
160 REM When you will RUN it, nothing will
170 REM happen, because REM does not instruct
180 REM the computer to do any thing.
190 REM
200 REM
210 END
RUN
```

বলাই বাহুল্য, প্রোগ্রামটি RUN করলে কিছুই ঘটবে না কারণ REM স্টেটমেন্ট থেকে CPU কোনো কাজ করার নির্দেশই পাবে না।

১২

REM

কম্পিউটারের সাহায্যে পড়ন্ত বস্তুর গতিবেগ বার করা (REM-এর প্রয়োগ)

আমরা যারাই বিজ্ঞান পড়েছি তারা Dynamics-এর এই প্রাথমিক ফর্মুলাটি জানি

$v^2 = u^2 + 2fS$

আর যাঁরা এই ফর্মুলাটি জানেন না, তাঁদের বলি, এটা একটি খুব কাজের ফর্মুলা। এর সাহায্যে কোনো ছুটন্ত বস্তুর প্রাথমিক গতিবেগ (Initial velccity), চূড়ান্ত গতিবেগ (Final velocity), সরণ (Displacement) এবং ত্বরণ (Accelaration). বার করা যায়।

ফর্মুলাতে যে সংকেতগুলি ব্যবহার করা হয়েছে তাদের অর্থ হলো

u = প্রাথমিক গতিবেগ (Initial velocity), (মাপা হবে ফুট প্রতি সেকেণ্ডে)

v = চূড়ান্ত গতিবেগ (Final velocity)

f = ত্বরণ (Accelaration), (মাপা হবে ফুট প্রতি সেকেণ্ড^2)

S = সরণ (Displacement), (মাপা হবে ফুটে)

[এই ফর্মুলায় আমরা F.P.S একক ব্যবহার করেছি, তাই দৈর্ঘ্য মাপতে ফুট ব্যবহার করা হয়েছে]

এই একই ফর্মুলার সাহায্যে উঁচু জায়গা থেকে ফেলে দেওয়া বস্তুর গতিবেগও বার করা যায়। সেক্ষেত্রে বস্তুর f বা ত্বরণ, মাধ্যাকর্ষণজাত ত্বরণের (Accela ration due to Gravity) সমান হয়। বিজ্ঞানে এই ত্বরণকে লিখতে G হরফ ব্যবহার করা হয় এবং এই G-র মান হলো সেকেণ্ডে 32 ফুট/সেকেণ্ড^2।

এই ফর্মুলা ব্যবহার করে স্কুলের ছাত্রদের অনেক অঙ্ক কষতে হয়। আমরা এবার দেখবো সেই সব অঙ্ক ছোট BASIC প্রোগ্রামের সাহায্যে কি ভাবে কম্পিউটারের মাধ্যমে করা যায়। এই প্রোগ্রামে আমাদের আগে শেখা সব কটি স্টেটমেন্টের ব্যবহারই দেখতে পাবো।

52

REM

```
NEW
10 REM Program to calculate the final velocity of a free falling body
20 REM Initial velocity=U, measured in feet/sec.
30 REM Final velocity=V, measured in feet/sec.
40 REM Accelaration due to gravity=G, measured in feet/sec., square.
50 REM Height of fall=H, measured in feet
60 REM Formulae used : VΛ2=UΛ2+2*G*H
70 PRINT "Type the values of U & H after the question mark"
80 INPUT U, H
90 LET G=32
100 LET V=SQR ( UΛ2+2*G*H)
110 PRINT "Velocity=", V, "feet/sec."
120 END
RUN
Type the values of U & H after the question mark
? 0, 57
Velocity=60·39867548 feet/sec.
```

আর একটি অঙ্ক করা যাক,

যেখানে U=7 এবং H=79·8976

সুতরাং আবার RUN টিপুন—

```
RUN
Type the values of U & H after the question mark.
? 7, 79·8976
Velocity=71·85016632 feet/sec.
```

REM

এই প্রোগ্রামের 10 থেকে 60 নম্বর লাইন পর্যন্ত কম্পিউটার কোনো নির্দেশ পাচ্ছেনা। ওই কটি লাইন কেবলমাত্র কম্পিউটার ব্যবহারকারীকে বুঝিয়ে দিচ্ছে এটি কিসের প্রোগ্রাম, এতে ব্যবহৃত বিভিন্ন নিউমেরিক ভেরিয়েবলগুলি কি কি এবং কি ফর্মূলা ব্যবহার করা হচ্ছে।

70 নম্বর লাইনে কম্পিউটারকে প্রশ্ন করার নির্দেশ দেওয়া হচ্ছে। 80 নম্বরে অজানা ভেরিয়েবলগুলির নাম কম্পিউটারকে জানানো হচ্ছে। 90 নম্বর লাইনে G নিউমেরিক ভেরিয়েবেলের একটি গাণিতিক মাণ assign অর্থাৎ নির্দিষ্ট করে দেওয়া হচ্ছে (আমরা জানি G=32 feet/sec², তাই LET G=32 লেখা হয়েছে)। 100 নম্বর লাইনে কম্পিউটারকে ফর্মূলাটি জানানো হচ্ছে। 110 নম্বর লাইনে উত্তর ছাপানোর নির্দেশ এবং 120 নম্বর লাইনে কম্পিউটারকে বলা হচ্ছে প্রোগ্রাম এখানেই শেষ হলো।

প্রোগ্রামটি দুবার RUN করে দুটি অঙ্ক কষা হয়েছে। প্রথম অঙ্কে ধরা হয়েছে বস্তুটি স্থির অবস্থা থেকে নিচে ফেলে দেওয়া হয়েছে এবং দ্বিতীয় অঙ্কে বস্তুটির প্রাথমিক কিছু গতিবেগ ছিলো ধরা হয়েছে।

এবার আসুন ধরা যাক, কম্পিউটারের সাহায্যে কত সহজে মেনসুরেশনের (Mensuration) অঙ্ক কষা যায়। উদাহরণ হিসেবে আমরা কিভাবে একটি বৃত্তের ক্ষেত্রফল নির্ণয়ের সফট্ওয়্যার লিখতে পারি সেটাই বিশ্লেষণ করা হল।

যে কোনো বৃত্তের ক্ষেত্রফল নির্ণয় করা

```
NEW
 1 REM It is a program to determine the area of
          a circle, whose radius is given.
 5 REM The formula used A=PI*(R∧2)
10 REM Where PI=3·1415927……
15 REM          A=area of the circle
20 REM          R=radius of the circle
25 PRINT "Type the unit of length you are
          using, after the question mark"
```

```
30 INPUT R $
35 PRINT "Type the length of the radius in" ;
         R $ ; "after the question mark"
40 INPUT R
50 LET   A=PI*(RΛ2)
60 PRINT
70 PRINT "The area of the circle is" ; A ;
         "Square" ; R $
80 END
RUN
Type the unit of length you are using, after the
question mark
? Centimeter
Type the length of the radius in centimeter afte
question mark
? 2
The area of the circle is 12·56637061 Square
centimeter.
```

ক্যাশমেমো তৈরীর সফট্‌ওয়্যার

কম্পিউটারের সাহায্যে ক্যাশমেমো তৈরী

শুধু অঙ্কই নয়, আমরা ইতিমধ্যে যে তিনটি দারুণ দরকারী কী ওয়ার্ড শিখেছি (PRINT, INPUT এবং LET), তার সাহায্যে দৈনন্দিন কাজের উপযোগী বহু ধরণের সফট্‌ওয়্যার আমরা লিখতে পারি। এর জন্য দরকার কেবলমাত্র বাঁধন ছাড়া কল্পনা শক্তি। যেমন ধরুন, স্বয়ংক্রিয় ক্যাশমেমো তৈরী করার প্রোগ্রাম লিখতে প্রধানত এই তিনটি কী ওয়ার্ডই যথেষ্ট। লক্ষ্য করবেন এই সফট্‌ওয়্যারে আমাদের ইতিপূর্বে শেখা সব কী ওয়ার্ডই ব্যবহার করা হয়েছে।

52

ক্যাশমেমো তৈরীর সফট্ওয়্যার

```
10 REM Program to prepare cash memo through
       COMPUTER
20 REM Prompt for the OPERATOR
30 PRINT "Type the name, address of the
       customer"
40 INPUT N$, A$
50 PRINT "Type the quantity, item, rate"
60 INPUT Q, I $, R
70 LET T=Q*R
80 LET S=INT (T*8)/100
90 LET N=T+S
100 REM Printing instruction
110 PRINT
120 PRINT "................................................"
130 PRINT
140 PRINT "CASH MEMO"
150 PRINT "...................."
160 PRINT
170 PRINT "Name :" ; N$
180 PRINT "Address :" ; A$
190 PRINT
200 PRINT
210 PRINT "Quantity purchased=" ; Q
220 PRINT
230 PRINT "Item purchased=" ; I$
240 PRINT
250 PRINT "Rate=" ; R ; "Rs"
260 PRINT
270 PRINT "Sales Tax 8%=" ; S ; "Rs"
280 PRINT
```

```
290 PRINT "... ....  ................................"
300 PRINT
310 PRINT "Total amount=" ; N ; "Rs"
320 PRINT
330 PRINT "......... ..................... .............."
340 PRINT
350 PRINT
360 END
RUN
Type the name, address of the customer
? Soumya Mitra, 15/2 Rani Sankari Lane
Calcutta 26
Type the quantity, item, rate
? 5, Apple II Computer, 10000.59
............ ...................................
          CASH MEMO
          ....................
Name : Soumya Mitra
Address : 15/2 Rani Sankari  Lane      Calcutta 26
Quantity Purchased=5
Item purchased=Apple II Computer
Rate=10000.59 Rs
Sales Tax 8%=4000.23 Rs.
..................................................
Total amount=54003.18 Rs
..................................................
```

এই প্রোগ্রামে সেলস্ ট্যাক্সের হার 8% হিসাবে ধরা হয়েছে। যদি সেলস্ ট্যাক্সের পরিমাণ অন্য হয় তবে লাইন নাম্বার 80তে লেখা স্টেটমেন্টটি পরিবর্তন করে নিতে হবে।

১২ IF THEN

IF THEN

দৈনন্দিন জীবনে আমরা প্রায়ই অবস্থা বুঝে ব্যবস্থা করি, ঠিক তেমনি, কম্পিউটার প্রোগ্রামেও অনেক সময় INPUT ভেরিয়েবল-এর মানের উপর নির্ভর করে কম্পিউটারকে চালনা করতে হয়। BASIC ভাষায় এই কাজের জন্য দুটি যুগ্ম স্টেটমেন্ট ব্যবহার করা হয়। স্টেটমেন্ট দুটি হলো IF এবং THEN.

প্রোগ্রামের একই লাইনে IF এবং THEN লেখা হয় যেমন—

50 IF A=64 THEN 200

প্রোগ্রামের 50 নম্বর লাইনে লেখা IF, THEN স্টেটমেন্টের অর্থ হলো—

যদি INPUT A, 64র সমান হয় তবে C.P.U যেন 50 নম্বর লাইনের পরবর্তী সমস্ত লাইনের নির্দেশ অগ্রাহ্য করে সরাসরি লাইন নাম্বার 200-এ চলে যায় এবং সেই লাইন থেকে সঠিক ক্রমে নির্দেশ পালন করে। আর যদি A, 64র থেকে ছোট বা বড় হয় তবে C.P.U যেন 50 নম্বরের পরবর্তী লাইনের নির্দেশগুলি পর পর সঠিক ক্রমে পালন করে যায়। ঠিক তেমনি 90 IF B$="B" THEN 110

নির্দেশটির অর্থ—

ইনপুট স্ট্রিং ভেরিয়েবল যদি B হরফ হয়, তবে C.P.U যেন 90-এর পরবর্তী লাইনের নির্দেশ অগ্রাহ্য করে সরাসরী 110 লাইন থেকে সঠিক ক্রমে নির্দেশ পালন করে। আর যদি ইনপুট B হরফের বদলে অন্য যে কোন হরফ হয়, তবে C.P.U যেন 90-এর ঠিক পরবর্তী লাইন থেকে নির্দেশ পালন করতে শুরু করে।

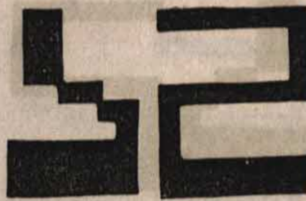
একটা ব্যাপার শুধু মনে রাখতে হবে, স্ট্রিং ভেরিয়েবল-এর মান সব সময় " " চিহ্নে মধ্যে লিখতে হবে (যেমন B $="B"), নিউমেরিক ভেরিয়েবলের ক্ষেত্রে কিন্তু এই নিয়ম নেই।

সফট্ওয়্যার রচনার ক্ষেত্রে IF THEN অত্যন্ত দরকারী

LET স্টেটমেন্ট ব্যবহার করে নিউমেরিক ভেরিয়েবলের মান নির্দিষ্ট করার নিয়ম আমরা ১৭০ পাতায় শিখেছি। স্ট্রিং ভেরিয়েবলের মান নির্দিষ্ট করার নিয়ম কিন্তু একটু আলাদা। এক্ষেত্রে যদি স্ট্রিং ভেরিয়েবলটি হয় A$ এবং তার মান হিসেবে BASIC কথাটি নির্দিষ্ট করতে চাই তবে BASIC কথাটি কোটেশন চিহ্নের মধ্যে লিখতে হবে। অর্থাৎ ব্যাপারটি হবে এই রকম :—

A$ = "BASIC"

স্টেটমেন্ট। IF THEN-এর সাহায্যে প্রোগ্রামার তার ইচ্ছামতো প্রোগ্রামকে একাধিক পথে চালিত করতে পারেন। একে কম্পিউটার পরিভাষায় বলা হয় ব্রাঞ্চিং (Branching)। তাছাড়া এই স্টেটমেন্টকে প্রোগ্রামার ছাঁকনির মতোও ব্যবহার করতে পারেন। IF THEN ছাঁকনির সাহায্যে অনায়াসে বিভিন্ন ধরনের ভেরিয়েবল্‌কে আলাদা করে ফেলা যায়। পরবর্তী সফট্‌ওয়্যারে IF THENকে ঠিক এই রকম ভাবে ব্যবহার করা হয়েছে।



IF THEN

কম্পিউটারের সাহায্যে বয়স অনুযায়ী অ্যাপ্লিকেশন ফর্ম বাছাই।

চাকরীর আবেদনপত্রে আবেদনকারীর বয়স সীমা (কিম্বা শিক্ষাগত যোগ্যতা) বেঁধে দেওয়া হয়। তবু বহু আবেদন পত্রে দেখা যায় আবেদনকারীর বয়স সীমা (অথবা যোগ্যতা) নির্ধারিত সীমার থেকে কম কিম্বা বেশী। এই ধরনের ফর্ম হাতে করে বাছাই করা যথেষ্ট সময় সাপেক্ষ ব্যাপার। অথচ IF THEN স্টেটমেন্ট ব্যবহার করে এই ফর্ম কম্পিউটারের সাহায্যে অত্যন্ত দ্রুত এবং নির্ভুলভাবে বাছাই করা সম্ভব।

ধরা যাক কোনো চাকরির ক্ষেত্রে বয়স সীমা বেঁধে দেওয়া হয়েছে 25 থেকে 30 বছর। এই চাকরির জন্য আবেদনপত্র যাচাই করার প্রোগ্রামটি হবে এই রকম—

```
NEW
10 REM Computerized   Application   Form
       Shorting
20 REM Eligible age limit 25 to 30 years
30 PRINT "Age" ;
40 INPUT A
50 IF A<25 THEN 90
60 IF A>30 THEN 90
70 PRINT "Form is eligible, can be sent for
          further processing"
80 STOP
90 PRINT "Reject"
```

১২

IF THEN

```
100 END
RUN
Age ? 18
Reject
RUN
Age ? 32
Reject
RUN
Age ? 26
Form is eligible, can be sent for further processing.
```

প্রোগ্রামটিতে ফর্ম বাছাইয়ের কাজ করা হচ্ছে স্টেটমেণ্ট নাম্বার 50 এবং 60-এর মাধ্যমে। ওই দুটি লাইনে IF, THEN কী ওয়ার্ডের সাহায্যে C.P.U-কে নির্দেশ দেওয়া হচ্ছে যদি নিউমেরিক ইনপুট ভেরিয়েবল Aর মান 25-এর কম কিম্বা 30-এর বেশী হয় তবে সোজা 90 নম্বর লাইনে চলে যেতে হবে। নতুবা 50 এবং 60-এর ঠিক পরবর্তী স্টেটমেণ্ট নাম্বার 70-এ যেতে হবে।

তাই প্রার্থীর বয়স নির্ধারিত সীমা অতিক্রম করলেই C.P.U সোজা চলে যাবে 90 লাইনে। সেখানের নির্দেশ অনুসারে কম্পিউটার ফর্মটিকে বাতিল বলে ঘোষণা করবে। আর যদি বয়স সঠিক সীমার মধ্যে থাকে তবে 70 নম্বর লাইনের নির্দেশ অনুযায়ী তাকে পরবর্তী যাচাইয়ের জন্য প্রোগ্রামারের কাছে পাঠিয়ে দেবে।

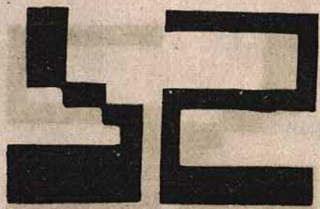
কম্পিউটারের সাহায্যে ফর্ম বাছাই বা পরীক্ষার খাতা দেখা আজকাল অত্যন্ত সাধারণ ঘটনা। কিন্তু মাঝে মধ্যে এই ধরনের ফলাফলে ত্রুটি ধরা পড়ে। সাধারণ মানুষের ধারণা এর জন্য বোধ হয় কম্পিউটারই দায়ী। অথচ ব্যাপারটি সম্পূর্ণ অন্য রকম। কম্পিউটারে পরীক্ষার খাতার নম্বর প্রোগ্রাম করতে গিয়ে মাঝে মধ্যে প্রোগ্রামার ভুল করে ফেলেন আর এর থেকেই কম্পিউটারের ফলাফলে ত্রুটি দেখা দেয়।

যদিও এই প্রোগ্রামে আমাদের বয়সটি হাতে টাইপ করতে হচ্ছে, কিন্তু বড় কম্পিউটারে এই কাজে O.C.R (Optical Character Recogniser) ইনপুট ডিভাইস ব্যবহার করা হয়। ফলে প্রোগ্রামের বয়সটি আর হাতে টাইপ করতে হয়না, পুরো কাজটিই সম্পূর্ণ স্বয়ংক্রিয় ভাবে কম্পিউটারই করে নেয়।

ইদানিং বহু কম্পিউটারে THEN স্টেটমেণ্ট লেখার পর PRINT স্টেটমেণ্টও সরাসরি লেখা যায় যেমন—

```
35 ... ... ...
45 IF A≥5 THEN PRINT "Game is over !"
55 ... ... ...
```

এই স্টেটমেন্টে কম্পিউটারকে নির্দেশ দেওয়া হচ্ছে, যদি নিউমেরিক ভেরিয়েবল A-এর মান 5-এর সমান কিম্বা বেশী হয় তবে V.D.Uতে 'Game is over !' কথাটি ফুটিয়ে তোল। আর যদি A-এর মান 5-এর থেকে কম হয় তবে 55 নম্বর এবং তার পরিবর্তিত স্টেটমেন্টের নির্দেশ পরপর পালন করে যাও।



IF THEN

IF, THEN ব্যবহার করে একটি ছোট্ট User friendly প্রোগ্রাম

```
NEW
1 REM User friendly software to introduce a
        freasher to a microcomputer.
10 PRINT "Hi !! I am your Microfriend ! What
        is your name" ;
20 INPUT N $
30 PRINT
40 PRINT
50 PRINT "Hello" ; N $; "I am very glad to
        meet you.  Do you like me" ;
60 INPUT A $
70 PRINT
80 PRINT
90 IF A$="NO" THEN 120
100 PRINT "I also like you" ; N $ ; "Please learn
        BASIC very quickly, so that we could
        chat freely.  Good bye" ; N $
110 STOP
120 PRINT "How dare you ! ( Sob !! Sob !! ) I
        never met such a rude human like you,
        Get lost you uncultured brute !"
130 END
```

IF THEN

এবার দেখা যাক প্রোগ্রামটি RUN করলে কি হয়।

RUN

Hi !! I am your Microfriend ! What is your name ?

Ratan

Hello Ratan, I am very glad to meet you. Do you like me ?

YES

I also like you Ratan ! Please learn BASIC very quickly, so that we could chat freely. Good bye Ratan

ধরুন, এবার অন্য আর একজন আবার প্রোগ্রামটি RUN করলো এবং কম্পিউটারের প্রশ্নের উত্তরে NO টাইপ করলো। তাহলে কি কাণ্ড হবে সেটা দেখা যাক !

RUN

Hi !! I am your Microfriend ! What is your name ?

Nishikanta

Hello Nishikanta, I am very glad to meet you. Do you like me ?

NO

How dare you ! (Sob !! Sob !!) I never met such a rude human like you. Get lost you uncultured brute !

প্রোগ্রামের 10 নম্বর লাইনটি লক্ষ্য করুন। সাধারণ PRINT স্টেটমেন্টের লাইন যে ভাবে লেখা হয় ঠিক সেইভাবে PRINT স্টেটমেন্টের পর যে লেখাটি V.D.U-তে ফোটাতে হবে, সেটি " " চিহ্নের মধ্যে লেখা হয়েছে। তবে একটি বৈশিষ্ট্য এই লাইনে আছে, সেটি হলো, লাইনের একেবারে শেষে ' ; ' চিহ্ন ব্যবহার। সেমিকোলন চিহ্ন বসানোর জন্যই কম্পিউটার V.D.U স্ক্রিনে "Hi !!···name" লাইনটি ফুটিয়ে তোলার পর INPUT জানতে চাওয়ার ' ? ' চিহ্নটি

ঠিক তার গায়েই বসিয়েছে। তাই RUN করার পর স্ক্রিনে ফুটে উঠেছে—

Hi !! I am your Microfriend ! What is your name ?

যদি সেমিকোলন চিহ্ন না বসানো হতো, তবে কম্পিউটার পরিবর্তী লাইনে '?' চিহ্নটি ফুটিয়ে তুলতো।

অর্থাৎ সেক্ষেত্রে V.D.U-তে দেখা যেতো

Hi !! I am your Microfriend ! What is your name

?

এই একই কায়দায় 50 নম্বর লাইনের স্টেটমেন্টটিও লেখা হয়েছে।

IF THEN

আরো একটি উদাহরণ

IF, THEN ব্যবহার করার সঠিক নিয়ম নিচের ছোট্ট প্রোগ্রাম থেকে খুব সহজেই বোঝা যাবে।

```
NEW
10 INPUT A
20 IF A<O THEN 40
30 PRINT A ; "is a positive number"
35 STOP
40 PRINT A ; "is a negative number"
50 END
RUN
? 10
10 is a positive number
RUN
? −45
−45 is a negative number
```

প্রোগ্রামের 20 নম্বর লাইনে IF ব্যবহার THEN করা হয়েছে। কী ওয়ার্ড দুটির মাধ্যমে কম্পিউটারকে নির্দেশ দেওয়া হয়েছে যে ইনপুট

নিউমেরিক ভেরিয়েবল্ A-র মান যদি 0-র থেকে কম হয়তবে স্টেটমেন্ট নম্বর 40-এ চলে যাও। নইলে 20 লাইনের ঠিক পরবর্তী লাইন অর্থাৎ 30 নম্বর লাইনের নির্দেশ পালন কর।

উদাহরণটি থেকে বুঝতেই পারছেন, IF THEN স্টেটমেন্ট ব্যবহার করে ভেরিয়েবলের বিশেষ মানের ওপর ভিত্তি করে আমরা প্রোগ্রামকে দুটি ভিন্ন পথে নিয়ে যেতে পারি। একে কম্পিউটার পরিভাষায় বলা হয় ব্রাঞ্চিং (Branching) অথবা কণ্ডিশনাল জাম্প (Conditional Jump)।

GOTO

IF, THEN স্টেটমেন্টের সাহায্যে আমরা কোন ভেরিয়েবলের মানের ওপর নির্ভর করে C.P.U-এর নিয়ন্ত্রণকে প্রোগ্রামের বিভিন্ন লাইনে চালনা করতে পারি। কিন্তু প্রোগ্রাম করা কালীন এমন অনেক পরিস্থিতির উদ্ভব হতে পারে যখন C.P.Uকে সরাসরি প্রোগ্রামের এক অংশ থেকে অন্য অংশে পাঠিয়ে দিতে হয়। কম্পিউটার পরিভাষায় একে বলা হয় নন্ কণ্ডিশনাল জাম্প (Non Conditional Jump)। এই নির্দেশের জন্য BASIC-এ GOTO কী ওয়ার্ড ব্যবহার করা হয়।

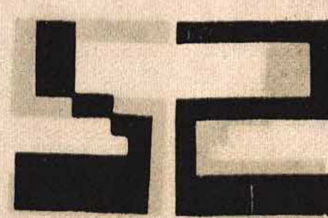
যেমন

```
30……
40 GOTO 90
50……
```

এই নির্দেশের অর্থ হলো, C.P.U যেই 40 নম্বর লাইনে আসবে, অমনি নিয়ন্ত্রণ পরবর্তী লাইন 50-এ না গিয়ে সরাসরি 90 নম্বর লাইনে চলে যাবে। লক্ষ্য করুন এক্ষেত্রে কিন্তু আমাদের C.P.U নিয়ন্ত্রণকে 90 নম্বর লাইনে পাঠাতে কোন ভেরিয়েবলের বিশেষ মানের ওপর নির্ভর করতে হচ্ছে না। তাই GOTO স্টেটমেন্টকে নন্ কণ্ডিশনাল জাম্প স্টেটমেন্ট বলা হয়।

GOTO স্টেটমেন্ট ব্যবহার করে আমরা C.P.U নিয়ন্ত্রণকে যে কোন লাইনেই পাঠাতে পারি। যদি GOTO-র সাহায্যে C.P.U নিয়ন্ত্রণকে পুনরায় GOTO স্টেটমেন্টের পূর্ববর্তী কোন স্টেটমেন্ট

লাইনে পাঠানো যায় তবে সৃষ্টি হয় একটি আবর্ত, যাকে কম্পিউটার পরিভাষায় বলা হয় লুপ (Loop)।



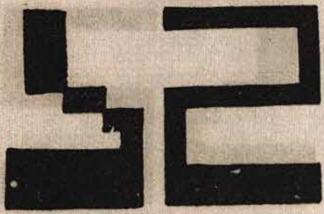
GOTO

উদাহরণ—

```
NEW
10 REM Demo program of GOTO Loop
20 REM An  endless  Loop  will  be  formed
                              between 50 and 60
30 PRINT "What is your name" ;
40 INPUT N $
50 PRINT N $ ;
60 GOTO 50
```

প্রোগ্রামটি RUN করলে কম্পিউটার 30 এবং 40 নম্বর লাইনে নির্দেশ অনুসারে আপনার নাম জানতে চাইবে। নাম টাইপ করে ENTER কিম্বা RETURN টিপলেই কম্পিউটার 50 নম্বর লাইনে দেওয়া নির্দেশ অনুসারে আপনার নাম V.D.U-তে ফুটিয়ে তুলবে। তারপর 60 নম্বরে লেখা GOTO (স্টেটমেন্টের ফলে পুনরায় 50 নম্বরে ফিরে যাবে এবং আবার আপনার নামটি V.D.U-তে লিখে দেবে। এই ভাবে 50 এবং 60 লাইনের মাঝে C.P.U-এর কন্ট্রোল ঘুরতেই থাকবে। ফলে আপনি দেখবেন ENTER কিম্বা RETURN টেপার পর সম্পূর্ণ V.D.U স্ক্রিন জুড়ে সারিবদ্ধ আপনার নাম ফুটে উঠলো।

এ ধরনের আবর্তের সঙ্গে সাইকেলের চেনের খুব মিল আছে। ঘুরন্ত সাইকেলের চেনের যেমন কোন শেষ নেই ঠিক তেমনি আবর্তটিও একবার চালু করলে চলতেই থাকবে, কখনো শেষ হবেনা (অবশ্য যদি না আপনি জোর করে কম্পিউটারকে থামিয়ে দেন অথবা সুইচ্ অফ্ করে দেন)। একে তাই অসীম আবর্ত বা Endless Loop বলা হয়। এণ্ড লেস্ লুপ আবর্ত কম্পিউটার প্রোগ্রামে সচরাচর ব্যবহার করা হয় না। কিন্তু এই আবর্তকেই সামান্য হেরফের করে কাজে লাগালে কয়েকটি ক্ষেত্রে খুব সুবিধা হয়। যেমন ধরুন IF, THEN ব্যবহার করে যে ফর্ম বাছাইয়ের প্রোগ্রামটি লেখা হয়েছিল, সেই প্রোগ্রামে এক একটি বয়স টাইপ করে ফলাফল জানার পর পুনরায় প্রোগ্রামটি চালু



GOTO

করার জন্য RUN টাইপ করতে হচ্ছিল। ব্যাপারটি বিরক্তিকর, কারণ আবেদন পত্রতো কেবল দু-তিনটি জমা পড়েনা। এই অসুবিধা খুব সহজেই একটি GOTO Jump এবং একটি endless loop ব্যবহার করে দূর করা যায়। এর জন্য প্রোগ্রামের 80 লাইনে STOP-এর বদলে GOTO 100 এবং শেষ লাইনে অর্থাৎ 100 নম্বর লাইনে END না বসিয়ে GOTO 10 লিখে দিলেই কাজ হয়ে যায়। তাহলে পুরো প্রোগ্রামটি দাঁড়ায় এই রকম—

```
10 REM Computerized Application Form
       Shorting
20 REM Eligible age limit 25 to 30 years
30 PRINT "Age";
40 INPUT A
50 IF A<25 THEN 90
60 IF A>30 THEN 90
70 PRINT "Form is eligible, Can be sent for
       further processing"
80 GOTO 100
90 PRINT "Reject"
100 GOTO 10
RUN
Age ? 18
Reject
Age ? 32
Reject
Age ? 26
Form is eligible, Can be sent for further
       processing
Age ?
```

অসীম আবর্ত থেকে প্রোগ্রামকে উদ্ধার করা

নিশ্চয়ই লক্ষ্য করেছেন ওপরের প্রোগ্রামটি একবার RUN করলে, কম্পিউটার উত্তর বলে দেবার পর কেবলই প্রার্থীদের বয়স জানতে

চাইছে। এখন প্রশ্ন হলো শেষ প্রার্থীর বয়স টাইপ করা হয়ে গেলেও কম্পিউটার যখন আপনার কাছে আরও বয়স জানতে চাইবে তখন আপনি কি করবেন? একমাত্র সমাধান হল বিরক্ত হয়ে কম্পিউটারের প্লাগটা টান মেরে সকেট থেকে খুলে দেওয়া! অথচ সামান্য একটু বুদ্ধি খাটিয়ে সফট্ওয়্যারটি লিখলেই এই পরিস্থিতি থেকে অনায়াসে বেরিয়ে আসা যায়। উপায়টি হলো Endless loop-এর ভিতর একটি IF, THEN স্টেটমেন্ট ঢুকিয়ে দেওয়া। আমরা জানি IF THEN স্টেটমেন্টের সাহায্যে কোন একটি ভেরিয়েবলের মানের ওপর নির্ভর করে C.P.U কন্ট্রোলকে যে কোন লাইনে নিয়ে যাওয়া যায়। IF, THEN-এর এই বিশেষ ক্ষমতাকে কাজে লাগালেই আমরা অনায়াসে যখন খুশি C.P.U নিয়ন্ত্রণকে অসীম আবর্ত থেকে বাইরে বার করে আনতে পারি।

এক্ষেত্রে সব থেকে সহজ উপায় হলো ইনপুট ভেরিয়েবল A অর্থাৎ প্রার্থীর বয়সকেই কাজে লাগানো। কোন প্রার্থীর বয়স শূন্য কিম্বা ঋণাত্মক বছর হতে পারেনা। এই সত্যকে কাজে লাগিয়ে যদি আমরা প্রোগ্রামে আর দুটি বাড়তি লাইন ঢুকিয়ে দিই যেমন লাইন 40 এবং 50-এর ফাঁকে 45 নম্বর লাইন সংযুক্ত করে তাতে লিখি।

45 IF A≤O THEN 110

এবং প্রোগ্রামের একদম শেষে আর একটি বাড়তি লাইন 110 সংযুক্ত করে তাতে লিখি,

110 END

তাহলে যখনই প্রোগ্রামটি শেষ করার ইচ্ছে হবে তখন কী বোর্ডে ইচ্ছাকৃতভাবে 0 কিম্বা যে কোন ঋণাত্মক বয়স টাইপ করলেই C.P.U নিয়ন্ত্রণ 45 লাইনে লেখা নির্দেশের প্রভাবে অসীম আবর্ত ছেড়ে 110 লাইনে চলে আসবে। সেই সঙ্গে প্রোগ্রামটিও সমাপ্ত হয়ে যাবে। সুতরাং পরিবর্তিত এবং সর্বাঙ্গ সুন্দর প্রোগ্রামটি দাঁড়াবে এই রকম—

```
10 REM Computerized Application Form
       Shorting
20 REM Eligible age limit 25 to 30 years
30 PRINT "Age" ;
40 INPUT A
```

```
45 IF A<=O THEN 110
50 IF A<25 THEN 90
60 IF A>30 THEN 90
70 PRINT "Form in eligible, can be sent for
        further processing"
80 GOTO 100
90 PRINT "Reject"
100 GOTO 10
110 END
RUN
Age ? 25
Form is eligible, can be send for further
        processing
Age ? 32
Reject
Age ? 0
```

আর একবার আমাদের শেখা সমস্ত স্টেটমেন্ট গুলির ব্যবহার ঝালিয়ে নেওয়া যাক। আসুন আমরা একটি বিশেষ ধরনের অঙ্ক কষার সফটওয়্যার লিখে ফেলি যাতে আমাদের ইতিপূর্বে শেখা সবকটি স্টেটমেন্টই ব্যবহার করা হয়েছে।

স্কুলের ছাত্রছাত্রীদের Factorial অংক কষা শেখানোর User friendly সফটওয়্যার।

এখনো পর্যন্ত আমরা যে কটি কী ওয়ার্ড শিখেছি, কেবল সেগুলি ব্যবহার করেই আমরা অনায়াসে চমৎকার সব প্রোগ্রাম লিখতে পারি। এই user friendly প্রোগ্রামটি তারই সুন্দর উদাহরণ। প্রোগ্রামটি এতো বেশী user friendly যে, কম্পিউটারে এটি RUN করার পর যে কোন কম্পিউটার সম্বন্ধে অনভিজ্ঞ ব্যক্তি এই প্রোগ্রামের মধ্যে ফ্যাকটোরিয়াল অঙ্ক কষা শিখতে পারবেন। কম্পিউটারই নিজে থেকে শিক্ষার্থীকে ক্রমাগত নানারকম প্রশ্ন করে ও নির্দেশ দিয়ে প্রায় জীবন্ত মানুষের মতই দেখিয়ে দেবে কি করে কম্পিউটারটি

চালাতে হবে। অর্থাৎ শিক্ষার্থীর কম্পিউটার প্রোগ্রামিং জানার বিন্দুমাত্র প্রয়োজনই হবে না।

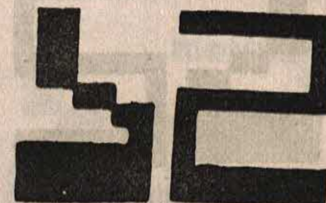
শুধুমাত্র অঙ্কই নয়, যে কোন শিক্ষণীয় বিষয়ই শিক্ষার্থীকে এমনভাবে কম্পিউটারের মাধ্যমে শেখানো সম্ভব।

```
10 PRINT "Hello friend, do you know what does
          the word FACTORIAL mean ? Please
          type YES or NO, and then press ENTER
          or RETURN Key"
20 INPUT  A$
30 IF A$="YES" THEN 180
40 PRINT "Never mind, I will tell you."
50 PRINT "FACTORIAL is a mathematical
          function."
60 PRINT "It is defined as :—"
70 PRINT "FACTORIAL  N=1*2*3*4*···*(N-2)*
          (N-1)*N"
80 PRINT "Where N is a POSITIVE INTEGER,
          in other words a positive whole number"
90 PRINT "Thus :—"
100 PRINT "FACTORIAL 1=1"
110 PRINT "FACTORIAL 2=2"
120 PRINT "FACTORIAL 3=1*2*3=6"
130 PRINT "FACTORIAL 4=1*2*3*4=24 & So
          on"
140 PRINT "The symbol of FACTORIAL is an
          exclamation mark !"
150 PRINT "That means ; FACTORIAL 4 is
          written as :— 4 !"
160 PRINT "FACTORIAL 0 is defined to be 1"
170 PRINT "Now you know all about FACTO-
          RIAL !"
```

52

GOTO

```
180 PRINT "OK then, let us do some FACTORIAL
          sum."
190 PRINT "You supply the number, and I will
          give the answer"
200 PRINT "BEFORE TYPING, PLEASE ENSURE
          THE NUMBER IS A POSITIVER
          INTEGER and after typing press ENTER
          or RETURN Key."
210 INPUT N
220 IF N=0 THEN 350
230 LET F=1
240 IF N<0 THEN 280
250 IF N<>INT (N) THEN 280
260 GOTO 300
270 PRINT
280 PRINT "Sorry friend, you are very careless, The
          number you typed, is not a positive
          integer"
290 GOTO 200
300 LET A=1
310 LET F=F*A
320 LET A=A+1
330 IF A=N+1 THEN 380
340 GOTO 310
350 PRINT
360 PRINT "Well, the answer is 1"
370 GOTO 400
380 PRINT
390 PRINT "Well, the answer is," ; N ; "1=" ; F
400 PRINT
```



ফ্যাকটেরিয়াল বার করার User friendly সফ্‌টওয়্যার

```
410 PRINT "(Type  YES or NO, then press ENTER
          or RETURN Key).  Want to caculate
          more" ;
420 INPUT B$
430 IF B$="YES" THEN 200
440 PRINT
450 PRINT "Good Bye !"
460 END
RUN
```

Hello friend, do you know what does the word
FACTORIAL mean ? Please type YES
or NO, and then. ENTER or RETURN
Rey.

? NO

Never mind, I will tell you.

FACTORIAL is a mathematical function.

It is defined as :—

FACTORIAL N=1*2*3*4*···*(N−2)*(N−1)*N

Where N is a POSITIVE INTEGER, in other words
a positive whole number Thus :—

FACTORIAL 1=1

FACTORIAL 2=2

FACTORIAL 3=1*2*3=6

FACTORIAL 4=1*2*3*4=24 so on

The symbol of FACTORIAL is an exclamation mark

That means, FACTORIAL 4 is written as :— 4 !

FACTORIAL 0 is defined to be 1

Now you know all about FACTORIAL

OK then, let us do some FACTORIAL sum.

You supply the number, and I will give the answer

১২

ফ্যাকটোরিয়াল বার করার User friendly সফট্ওয়্যার

```
BEFORE  TYPING, PLEASE  ENSURE  THE
        NUMBER IS A POSITIVE INTEGER,
        and after typing press ENTER or
        RETURN Key.
?—1
Sorry friend you are very careless.  The number you
        typed, is not a positive integer.
BEFORE  TYPING, PLEASE  ENSURE  THE
        NUMBER IS A POSITIVE INTEGER,
        and after typing press ENTER or
        RETURN Key.
?3
Well, the answer is, 3 !=6
(Type YES or NO, then press ENTER or RETURN
        Key).  Want to calculate more ? NO
Good Bye !
RUN
Hello friend, do you know what does the word
        FACTORIAL mean ?  Please type YES
        or NO, and the press ENTER or
        RETURN Key.
? YES
OK then, let us do some FACTORIAL sum.
You supply the number, and I will give the answer.
BEFORE  TYPING, PLEASE  ENSURE  THE
        NUMBER IS A  POSITIVE INTEGER,
        and after typing press ENTER or
        RETURN Key.
?0
```

১২

ফ্যাকটোরিয়াল বার করার User friendly সফট্ওয়্যার

Well, the answer is 1
(Type YES or NO, then press ENTER or RETURN
Key). Want to calculate more ? YES.
BEFORE TYPING PLEASE ENSURE THE
NUMBER IS A POSITIVE INTEGER,
and after typing press ENTER or
RETURN Key.
? 5
Well, the answer is, 5 != 120
(Type YES or NO, then press ENTER or RETURN
Key). Want to calculate more ? NO
Good Bye !

প্রোগ্রামের বিশ্লেষণ

আমাদের এই প্রোগ্রামের মূল অংশটি 210 থেকে 350 লাইনের মধ্যে লেখা আছে। অন্যান্য লাইনে প্রোগ্রামটি user friendly করে তোলার নির্দেশ দেওয়া হয়েছে। ইচ্ছা করলে কেবলমাত্র 210 থেকে 350 লাইনের স্টেটমেন্ট টাইপ করে প্রোগ্রামটি RUN করানো যেতে পারে। আসুন এবার আমরা 210 থেকে 350 লাইন অবধি প্রোগ্রামটিকে বিশ্লেষণ করে দেখি।

210 INPUT N

যে সংখ্যাটির ফ্যাকটোরিয়াল বার করতে হবে। সেই সংখ্যাটি জানতে চাওয়ার নির্দেশ দেওয়া হয়েছে।

220 IF N = 0 THEN 350

ফ্যাকটোরিয়ালের সংজ্ঞা অনুযায়ী 0 != 1, অথচ ফ্যাকটোরিয়ালের ফর্মুলায় সরাসরি 0 বসালে এই উত্তর পাওয়া যায় না। প্রোগ্রামার যদি কী বোর্ডে 0 টেপেন, তবে কম্পিউটার যাতে ভুল উত্তর না দেয় তারই ব্যবস্থা স্বরূপ এই লাইনে C.P.Uকে সরাসরি 350 লাইনে পাঠিয়ে দেওয়ার নির্দেশ দেওয়া হয়েছে। 350 এবং 360 লাইনে 0 ! এর উত্তর 1 ছাপানোর নির্দেশ দেওয়া আছে। সুতরাং কী বোর্ডে 0

১২

ফ্যাকটোরিয়াল বার করার
সফট্ওয়্যার বিশ্লেষণ

টিপলেই কম্পিউটার ফ্যাকটোরিয়ালের ফর্মুলায় অঙ্ক না কষে সরাসরি 360 লাইনে লেখা উত্তর বলে দেবে।

230 LET F=1

F হল একটি নিউমেরিক ভেরিয়েবল, যার মান 230 লাইনে LET স্টেটমেন্টের সাহায্যে 1 ঠিক করা হয়েছে।

240 IF N<O THEN 280

250 IF N< > INT (N) THEN 280

এই দুটি লাইনের মাধ্যমে ইনপুট ভেরিয়েবলটি Positive integer (ধণাত্মক পূর্ণ সংখ্যা) কিনা, তাই পরীক্ষা করা হচ্ছে। যদি সংখ্যাটি পজেটিভ ইন্টিজার না হয়, তবে C.P.U-এর নিয়ন্ত্রণ 280 লাইনে পাঠানো হচ্ছে। আর যদি সংখ্যাটি পজেটিভ ইন্টিজার হয় তবে C.P.U-এর নিয়ন্ত্রণ 260 লাইনে পাঠানো হচ্ছে।

260 GOTO 300

সংখ্যাটি যদি পজেটিভ ইন্টিজার হয়, তবে C.P.U-এর নিয়ন্ত্রণ 240 এবং 250 নম্বার লাইনের নির্দেশ পালন করে 260 লাইনে চলে আসবে। 260 লাইনে নিয়ন্ত্রণ আসা মাত্র C.P.U-এর নিয়ন্ত্রণকে সোজা 300 লাইনে অঙ্ক কষার জন্য পাঠিয়ে দেওয়া হচ্ছে।

270 PRINT

280 PRINT "Sorry friend, you are very careless. The number you typed, is not a positive integer."

প্রোগ্রামার যদি ভুল বশত কিম্বা ইচ্ছা করে Positive integer না টাইপ করেন, তবে C.P.U নিয়ন্ত্রণ 240, 250 লাইন থেকে সটান 270 লাইনে আসবে। 270 লাইনটি ছাপার সময় এক লাইন ফাঁক রাখবে। তারপর কন্ট্রোল যাবে 280 লাইনে। এই লাইনের মাধ্যমে প্রোগ্রামারকে মৃদু বকুনি দেওয়ার নির্দেশ দেওয়া হয়েছে।

290 GOTO 200

মনিটর স্ক্রিণের মাধ্যমে প্রোগ্রামারকে তিরস্কার করার পর প্রোগ্রামারকে পুনরায় নতুন সুযোগ দেওয়ার জন্য C.P.U নিয়ন্ত্রণকে 200 নম্বর লাইনে ফিরিয়ে নিয়ে যাওয়া হচ্ছে।

300 LET A=1

সঠিক সংখ্যা টাইপ করা হলে 260 লাইন থেকে নিয়ন্ত্রণ এই লাইনে চলে আসবে। এই লাইন থেকেই ফ্যাকটোরিয়ালের ফর্মুলা শুরু হচ্ছে। এখানে A একটি নিউমেরিক ভেরিয়েবল। LET স্টেটমেন্টের সাহায্যে তার প্রাথমিক মান 1 ঠিক করা হয়েছে।

310 LET F=F*A

এই লাইন থেকে হিসেব শুরু হচ্ছে। F ভেরিয়েবলের মান আমরা 230 লাইনে 1 নির্দিষ্ট (assign) করেছিলাম। এবার F-এর সেই মানের সঙ্গে Aর মান গুণ করা হচ্ছে এবং গুণফলকে নতুন F-এর মান বলে স্থির করা হচ্ছে। অর্থাৎ যে ঘটনাটি ঘটছে সেটি হলো :—

F=1*1=1

320 LET A=A+1

A-এর যা পূর্ববর্তী মান ছিলো, তার সঙ্গে 1 যোগ করে যে যোগফল হচ্ছে, সেটিই A-র নতুন মান বলে স্থির করা হচ্ছে। (এক্ষেত্রে Aর পূর্ববর্তী মান ছিলো 1, সুতরাং 320 লাইনে Aর নতুন মান হচ্ছে A=1+1=2)

330 IF A=N+1 THEN 380

330 লাইনে পরীক্ষা করা হচ্ছে আমাদের আকাঙ্ক্ষিত ফলাফল পাওয়া গেল কিনা। যদি নতুন Aর মান N+1-এর সমান না হয়, তবে নিয়ন্ত্রণ যাচ্ছে পরবর্তী লাইন 340-এ।

340 GOTO 310

340 লাইনে আসামাত্র পুনরায় হিসেব কষার জন্য নিয়ন্ত্রণকে 310 লাইনে পাঠানো হচ্ছে। সুতরাং 310 থেকে 340 লাইনের মধ্যে একটি আবর্ত সৃষ্টি হচ্ছে। যতক্ষণ পর্যন্ত না A-র নতুন মান N+1-এর সমান হয় ততক্ষণ এই আবর্তে চলতে থাকবে।

350 PRINT

360 PRINT "Well, the answer is 1"

370 GOTO 400

প্রোগ্রামার যদি 0-র ফ্যাকটোরিয়াল জানতে চান, তবে 220 লাইন থেকে নিয়ন্ত্রণ সোজা 350-এ আসবে। এই লাইনের PRINT স্টেটমেন্টের জন্য মনিটর স্ক্রিনে এক লাইন ফাঁক হবে। তারপর 360

লাইনে লেখা উত্তরটি ছাপা হবে। উত্তর ছাপার পর 370 লাইন অনুসারে নিয়ন্ত্রণ চলে যাবে 400 লাইনে।

380 PRINT

390 PRINT "Well, the answer is," ; "! = " ; F

হিসেব শেষ হলে C.P.U নিয়ন্ত্রণ 310 এবং 340-এর আবর্ত ছেড়ে 380 লাইনে চলে আসবে। এখানে আকাঙ্ক্ষিত উত্তরটি ছাপাবার নির্দেশ দেওয়া আছে।

400 PRINT

⋮

এই লাইন থেকে পরবর্তী লাইনগুলিতে প্রোগ্রামটি user friendly করে তোলার উপযুক্ত সমস্ত নির্দেশ দেওয়া আছে।

আবর্তের সাহায্যে কিভাবে অঙ্কটি কষা হচ্ছে

আমরা জেনেছি $N\,! = 1*2*3*4*\cdots*(N-1)*N$

এই ফর্মুলাটিই 300 থেকে 340-এর আবর্তে সৃষ্টি করা হচ্ছে।

প্রথমে F-এর প্রাথমিক মান ধরা হচ্ছে 1 এবং A-এর প্রাথমিক মান ধরা হচ্ছে 1

ধরা যাক আমরা $N=4$ টাইপ করেছি। সুতরাং প্রথমবার যখন C.P.U ওই আবর্তের মধ্যে দিয়ে যাবে তখন যা ঘটবে সেটি হলো—

F-এর বর্তমান মান = F-এর পূর্বের মান * A-এর পূর্বের মান

$= 1*1 = 1$

∴ F-এর বর্তমান মান 1

এবং A-এর বর্তমান মান = A-র পূর্বের মান + 1

$= 1+1 = 2$

যেহেতু $2 \neq 4+1=5$

তাই $A \neq N+1$

সুতরাং C.P.U আবার নতুন করে আবর্তে ঘুরবে। দ্বিতীয় আবর্তের সময় F-এর বর্তমান মান = F-এর পূর্বের মান * A-এর পূর্বের মান

$= 1*2$

ফ্যাকটোরিয়াল বার করার সফ্‌টওয়্যার বিশ্লেষণ

এবং A-এর বর্তমান মান = A-এর পূর্বের মান + 1

$= 2 + 1 = 3$

যেহেতু $3 \neq 4 + 1 = 5$

তাই $A \neq N + 1$

সুতরাং আবার C.P.U আবর্তের গোড়ায় ফিরে যাবে। এবার F এবং A-র মানগুলি হবে :—

F-এর বর্তমান মান = F-এর পূর্বের মান * A-র পূর্বের মান

= 1*2*3

এবং A-র বর্তমান মান = A-র পূর্বের মান + 1

$= 3 + 1 = 4$

যেহেতু $4 \neq 4 + 1 = 5$

তাই $A \neq N + 1$

তাহলে C.P.U-কে আর একবার আবর্তে ঘুরতে হবে। এই শেষ আবর্তে ঘোরার সময় F এবং A-র মান হবে :—

F-এর বর্তমান মান = F-এর পূর্বের মান * A-র পূর্বের মান

= 1*2*3*4

এবং A-র বর্তমান মান = A-র পূর্বের মান + 1

$= 4 + 1 = 5$

এবার $5 = 4 + 1$

তাহলে F-এর এই সর্বশেষ মানটিই হবে 4 !-এর মান।

মিলিয়ে দেখুন 4 ! = 1*2*3*4

এবং F = 1*2*3*4

সুতরাং F = 4 !

শুধু ফ্যাকটোরিয়ালই নয়, এইভাবে অন্য যে কোনো অঙ্ক আমরা কম্পিউটারের সাহায্যে কষতে পারি।

১২ FOR TO NEXT

FOR, TO, NEXT

GOTO আবর্ত ব্যবহারের প্রধান অসুবিধা হলো আবর্ত থেকে C.P.U নিয়ন্ত্রণকে বার করে আনতে IF THEN ব্যবহার করতে হয়। কেবলমাত্র আবর্ত তৈরী করার জন্যই BASIC ভাষায় বিশেষ তিনটি স্টেটমেন্টের ব্যবস্থা রাখা হয়েছে। এরা হলো FOR, TO এবং NEXT. FOR, TO একই লাইনে লিখতে হয় (যেমন IF. THEN এক লাইনে লেখা হয়) আর আবর্ত যে লাইনে শেষ হচ্ছে ; সেই লাইনে NEXT স্টেটমেন্ট দিতে হয়। নিচে একটি ছোট FOR, TO, NEXT আবর্তের প্রোগ্রাম দেওয়া হলো।

```
   10FOR A=1 TO 10
   20PRINT "SOUMYA MITRA"
   30NEXT A
>RUN
SOUMYA MITRA
SOUMYA MITRA
SOUMYA MITRA
SOUMYA MITRA
SOUMYA MITRA
SOUMYA MITRA
SOUMYA MITRA
SOUMYA MITRA
SOUMYA MITRA
SOUMYA MITRA
```

প্রোগ্রামটি RUN করলে, SOUMYA MITRA নামটি দশবার ছাপা হবে। এই প্রোগ্রামের 10 নম্বর লাইনে লেখা হয়েছে

10 FOR A=1 TO 10

নিউমেরিক ভেরিয়েবল Aকে এখানে Loop Counter বা আবর্ত

52

FOR, TO, NEXT

গণক হিসেবে ব্যবহার করা হয়েছে। A-র প্রাথমিক মান ঠিক করা হয়েছে 1, FOR, TO থেকে NEXT, প্রতি আবর্ত ঘোরার পর A-র মান এক করে বাড়তে থাকবে। এইভাবে A-র মান 1 থেকে 10 হলেই C.P.U কণ্ট্রোল আবর্ত ছেড়ে বেরিয়ে আসবে। আবর্তটি শুরু হবে লাইন নাম্বার 10 থেকে। C.P.U 10, 20 লাইনের নির্দেশ পালন করে 30 নাম্বার লাইনে এলেই NEXT A স্টেটমেন্ট তাকে ফের 10 নাম্বার লাইনে পাঠিয়ে দেবে। সুতরাং FOR, TO এবং NEXT লেখা স্টেটমেন্ট দুটির মধ্যে আবর্তটি কেবলমাত্র দশ পাকের জন্য আবদ্ধ থাকবে।

FOR, TO, NEXT ব্যবহার করে সেলসিয়াস থেকে ফারেনহাইট সারণী তৈরী করা

```
10 REM Program to convert celsius to Farenheit
20 PRINT "Celsius", "Farenheit"
30 LET C=0
50 FOR A=1 TO 11
55 LET F=(C*9/5)+32
60 PRINT C, F
70 LET C=C+10
80 NEXT A
90 END
RUN
```

Celsius	Farenheit
0	32
10	50
20	68
30	86
40	104
50	122

FOR, TO, NEXT

60	140
70	158
80	176
90	194
100	212

FOR, NEXT আবর্তের সাহায্যে 1 থেকে 10 অবধি সংখ্যার যোগফল নির্ণয়

```
10 LET X=0
20 FOR A=1 TO 10
30 LET X=X+A
40 PRINT A
50 NEXT A
60 PRINT "—"
70 PRINT X
80 END
RUN
      1
      2
      3
      4
      5
      6
      7
      8
      9
     10
    ————
     55
```

এই একইভাবে আমরা FOR NEXT আবর্তের সাহায্যে A. P. Series-এর যোগফল বার করতে পারি। সেক্ষেত্রে আমাদের 10 নম্বর

১২

FOR, TO, NEXT

লাইনে X-এর মান 0-র বদলে A. P. সিরিজের 1st term-এর সমান করতে হবে। এবং Common difference-এর ব্যাপারটি 20 নাম্বার লাইনে ঢোকাতে হবে।

ধরা যাক আমরা নিচের সিরিজের যোগফল বার করবো।

$$5+7+9+11+13+15=?$$

এই A. P. সিরিজে 6টি পদ বা term আছে। সিরিজের প্রথম termটি হলো 5 এবং পর পর দুটি পদের ব্যবধান হলো 2. সুতরাং এই সিরিজের যোগফল বার করতে আমাদের 6 বারের একটি আবর্ত সৃষ্টি করতে হবে যাতে Loop Counter-এর প্রাথমিক মান হবে 5 এবং প্রতি আবর্তে সেটি 2 করে বাড়বে। এইটা করতে আমাদের লিখতে হবে—

```
10 LET X=O
20 FOR A=5 TO 15 STEP 2
30 LET X=X+A
40 PRINT A
50 NEXT A
60 PRINT "—"
70 PRINT X
80 END
RUN
     5
     7
     9
    11
    13
    15
   ———
    60
```

20 নাম্বার লাইনে A=5 লেখার জন্য Loop Counter A-র প্রাথমিক মান 5 হয়েছে। STEP 2 লেখার ফলে প্রতি আবর্তে A-র মান 2 করে বাড়বে।

১২

FOR, TO, NEXT

FOR, NEXT আবর্তে Loop Counter-এর মান শুধু যে ক্রমাগত বাড়াতেই হবে এমন কোন নিয়ম কিন্তু নেই। প্রয়োজন হলে মানটি বেশী থেকে কমও করা যেতে পারে। সেক্ষেত্রে ঋণাত্মক STEP ব্যবহার করতে হবে। যেমন—

```
10 FOR A=10 TO 0 STEP —2
20 PRINT A
30 NEXT A
40 END
RUN
    10
     8
     6
     4
     2
```

FOR, NEXT আবর্তের সাহায্যে Time Delay সৃষ্টি

কম্পিউটার যখন লেখা ছাপায়, তখন V.D.U-তে লেখাগুলি খুব তাড়াতাড়ি ফুটে ওঠে। কিন্তু অনেক সময় স্ক্রিন ভর্তি লেখা চট করে ফুটে ওঠার থেকে যদি লেখাগুলি ধীরে ধীরে এক লাইন এক লাইন করে ফোটে, তবে পড়বার সুবিধা হয়। তাছাড়া বহু প্রোগ্রামে কোনো দুটি স্টেটমেন্টের নির্দেশ পালন করার মাঝে অনেক সময়ই কিছু সময় ব্যবধান রাখার প্রয়োজন হয়। এ ধরনের কাজে FOR, NEXT আবর্ত খুব কাজে দেয়। নিচে এ ধরনের একটি প্রোগ্রাম দেখানো হলো। এই প্রোগ্রামটি RUN করলে দেখা যাবে স্ক্রিনে একটি লাইন লেখা ফুটে ওঠার কয়েক সেকেণ্ড পর পরবর্তী লাইনটি ফুটে উঠবে। এই সময় ব্যবধানটিও FOR NEXT আবর্তের সাহায্যে খুব সহজেই নিয়ন্ত্রণ করা যাবে।

```
NEW
10 PRINT "It is a demonstration program of
         FOR NEXT Loop"
```



FOR, TO, NEXT

```
20 FOR X=1 TO 100
30 NEXT X
40 PRINT "This time delay is achive by the
        FOR, NEXT Loop".
50 FOR X=1 TO 100
60 NEXT X
70 PRINT "TO alter the time delay you have
        to alter the number after TO".
80 FOR X=1 TO 100
90 NEXT X
100 PRINT "This delay is achived with FOR X=
        1 TO 100
110 PRINT "Now lets see what happens with
        FOR X=1 TO 10"
120 FOR X=1 TO 10
130 NEXT X
140 PRINT "Now, with, FOR X=1 TO 1000"
150 FOR X=1 TO 1000
160 NEXT X
170 PRINT "I hope, you have understood the
        demonstration !"
180 FOR X=1 TO 100
190 NEXT X
200 PRINT "Good Bye"
210 END
```

TAB

সাধারণ PRINT কম্যাণ্ড / স্টেটমেন্টের সাহায্যে V.D.U-তে লেখা ফুটিয়ে তোলা খুব সহজ। কিন্তু লেখাটি V.D.U স্ক্রিনের ঠিক কোন অংশে ফুটে উঠবে তা আগে থেকে নিয়ন্ত্রণ করা যায় না।

52

TAB

তাই ফর্ম জাতীয় জিনিস কম্পিউটারের মাধ্যমে ছাপতে গেলে, সাধারণ PRINT স্টেটমেন্ট ব্যবহারে অসুবিধা দেখা দেয়। এক্ষেত্রে ব্যবহার করতে TAB স্টেটমেন্ট।

TAB কথাটি এসেছে tabulate থেকে। PRINT-এর সঙ্গে TAB স্টেটমেন্ট একত্রে বসিয়ে, আমরা খুব সহজেই বিষয় বস্তুটি লাইনের কোন জায়গায় ছাপা হবে, সেটি নিয়ন্ত্রণ করতে পারি। শুধু তাই নয়, একটু কায়দা করে ব্যবহার করতে পারলে, TAB স্টেটমেন্টের সাহায্যে অনেক গাণিতিক ফাংশনের গ্রাফও আঁকা যায়।

TAB স্টেটমেন্ট ব্যবহারের একটি সাধারণ উদাহরণ নিচে দেওয়া হলো—

```
40 PRINT A ; TAB (10) ; B ; TAB (15) ; C
```

এই স্টেটমেন্টে কম্পিউটারকে নির্দেশ দেওয়া হয়েছে—

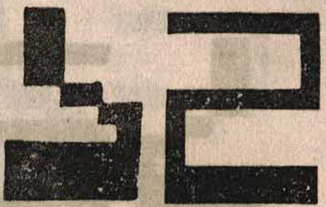
প্রথমে V.D.U বা প্রিন্টারের কাগজের বাঁ কোনে A-র মান ফুটিয়ে তোলো। তারপর 10 ঘর জায়গা ফাঁকা রেখে B-র মান লেখো। তারপর আবার 15 ঘর জায়গা ছেড়ে C-র মান লেখো।

উত্তর ছাপানোয় TAB-এর ব্যবহার

```
NEW
10 LET A=1
20 LET B=5·637
30 LET C= ·0039
40 PRINT TAB (5) ; A ; TAB (10) ; B :
      TAB (10) : C
50 END
RUN
      1          5·637          ·0039
```

গ্রাফ ছাপানোর TAB-এর ব্যবহার

TAB-এর সাহায্যে কেবলমাত্র বিষয়বস্তুটি কোন Column (স্তম্ভে)-এ ছাপা হবে, সেটি নিয়ন্ত্রণ করা যায়।



TAB

```
     10FOR A= 1 TO 10
     20PRINT TAB(A);A
     30NEXT A
     40END
>RUN
 1
  2
   3
    4
     5
      6
       7
        8
         9
          10
```

ওপরের প্রোগ্রামটি থেকে পরিষ্কার বোঝা যাচ্ছে কি ভাবে TAB-এর সাহায্য নিয়ে কোনো ক্যারেকটারকে ক্রমাগত ডানদিকে সরিয়ে সরিয়ে ছাপা যায়। এখানে TAB স্টেটমেন্টটির একটি FOR, TO, NEXT লুপের মধ্যে বসানো হয়েছে। FOR, NEXT লুপের প্রতি আবর্ত ঘোরার সঙ্গে সঙ্গে লুপ কাউন্টার ভেরিয়েবল A-র মান এক করে বাড়ছে এবং সেই মানটি প্রতি লাইনে ক্রমাগত এক লাইন করে ডানদিক সরিয়ে ছাপা হয়েছে।

```
     10FOR A= 1 TO 10
     20PRINT TAB(A);"*"
     30NEXT A
     40END
>RUN
 *
  *
   *
    *
     *
      *
       *
        *
         *
          *
```

এই প্রোগ্রামটি অনেকটা আগেরই মতো, তবে এখানে TAB-এর সাহায্যে লুপ কাউন্টার ভেরিয়েবলের মান না ছাপিয়ে '*' ক্যারেকটারটি ছাপা হয়েছে। এই ভাবে ছাপার ফলে একটি বাঁকা সরলরেখার সৃষ্টি হয়েছে। এ ধরনের প্রোগ্রামকে আর একটু অদল বদল করে নিয়ে আমরা অনায়াসে বহু গাণিতিক ফাংশানের গ্রাফও আঁকতে পারি। পরবর্তী প্রোগ্রামে এই ধরনেরই একটি প্রোগ্রাম দেখানো হয়েছে।

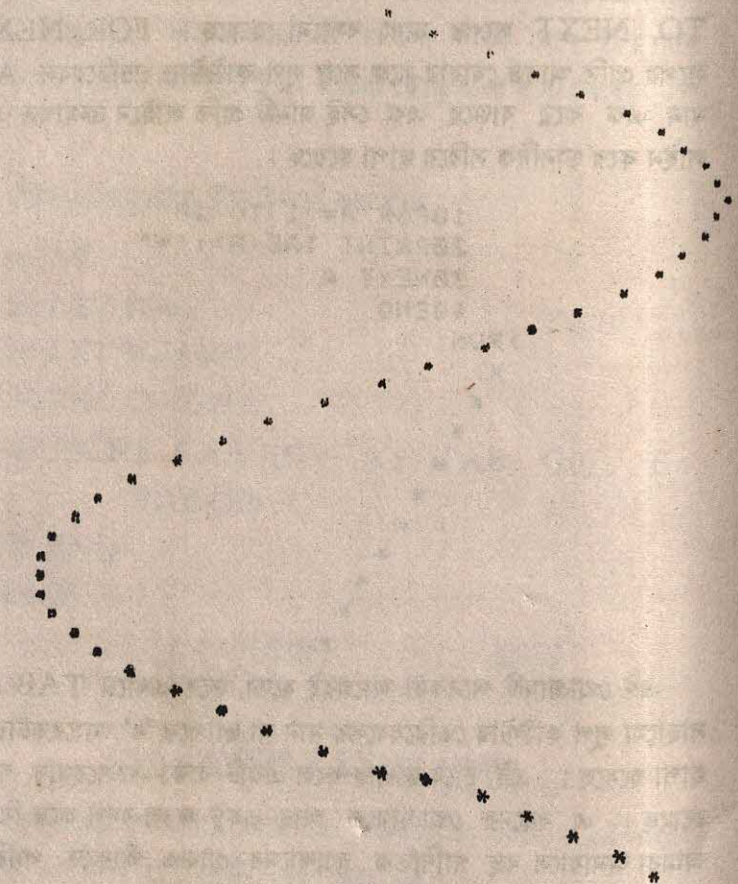
১২

TAB

TAB ব্যবহার করে আমরা কোনো ক্যারেকটারকে স্ক্রিনের বা কাগজের মধ্যে পাশাপাশি সরাতে পারি। অপরপক্ষে কম্পিউটার প্রিন্টার প্রতি লাইনে লেখা ছাপার পর তার নিচের লাইনে লেখা ছাপে। এই ব্যবস্থার ফলে, TAB স্টেটমেন্টের সাহায্যে গ্রাফ আঁকা খুব সহজ হয়ে যায়। ওপাতায় আমরা দেখেছি, TAB-এর সাহায্যে প্রতি লাইন ছাপার সময় কোনো ক্যারেকটারকে ক্রমাগত ডান দিকে সরিয়ে আনলে স্ক্রিনে বা কাগজে কেমন একটি বাঁকা রেখার সৃষ্টি হয়।

এই বৈশিষ্ট্যকে কাজে লাগিয়ে Sinusoidal তরঙ্গের একটি গ্রাফ আঁকা হয়েছে।

```
10LET X=0
20FOR A=1 TO 50
30PRINT TAB(SIN(X)*30+31);"*"
40LET X=X+3.14159265/20
50NEXT A
60END
>RUN
```



এইভাবে গ্রাফ আঁকার মূল কায়দাটি হলো, TAB-এর সাহায্যে '*' চিহ্নটিকে Sin function অনুসারে ক্রমাগত স্ক্রিনের এপাশ-ওপাশ সরানো (যেহেতু প্রতিটি '*' চিহ্ন ছাপার পর, পরবর্তী '*' চিহ্নটি তার নিচের লাইনে ছাপা হচ্ছে, তাই স্ক্রিনে বা কাগজে, সময়ের সঙ্গে '*' চিহ্নটির সরণ বা Displacement-এর গ্রাফ আঁকা হয়ে যাচ্ছে)।

এটি খুব সহজেই করা যায়, যদি আমরা কী বোর্ডে টাইপ করি

30 PRINT TAB (Sin (X)) ; "*"

কিন্তু যেহেতু Sin X-এর সর্বাধিক এবং সর্বনিম্ন মান মাত্র 1 এবং −1, তাই এই নির্দেশ দিলে '*' চিহ্নটি স্ক্রিনের মধ্যে মাত্র তিন ঘর এপাশ-ওপাশ সরবে। অবশ্য সেটিও বাস্তব ক্ষেত্রে সম্ভব হবেনা, কারণ BASIC ভাষায় TAB (−1)-এর কোন অস্তিত্বই নেই। সুতরাং '*'-এর এপাশ-ওপাশ সরণের (amplitude) পরিমাণ বাড়াতে (যাতে গ্রাফের আকৃতি ভাল করে বোঝা যায়) Sin (X)-এর সঙ্গে 30 গুণ করা হয়েছে। এর ফলে TAB (0) থেকে TAB (30) অবধি '*' ছাপা হবে কিন্তু TAB (−30) থেকে TAB (−1) পর্যন্ত গ্রাফের কিছুই দেখা যাবে না। তাই পরিশেষে গ্রাফটিকে স্ক্রিন বা কাগজের মাঝে আনতে (Sin (X)*30)-র সঙ্গে 31 যোগ করা হয়েছে। TAB (Sin (X)*30+31)-এর মান কখনোই TAB(1)-এর থেকে ছোট হবে না আর গ্রাফটিরও কোন অংশ মার্জিনের বাইরে কাটা পড়বে না।

সারসংক্ষেপ

TAB ব্যবহার করে গ্রাফ আঁকার সময় নিচে দেওয়া দুটি নিয়ম সব সময় মনে রাখবেন।

(1) যদি গ্রাফের amplitude X গুণ বাড়ানোর প্রয়োজন হয়, তবে TAB-এর অন্তর্গত গাণিতিক ফাংশনের সঙ্গে X গুণ করবেন।

(2) X গুণ গাণিতিক ফাংশনটির সর্ব নিম্ন মান যদি −ve (ঋণাত্মক) হয়, তবে TAB-এর অন্তর্গত X গুণ গাণিতিক ফাংশনের সঙ্গে (X গুণ গাণিতিক ফাংশনের সর্বনিম্ন মান +1) যোগ করবেন।

১২

TAB

TAB এবং FOR, NEXT ব্যবহার করে হ্যাণ্ডবিল ছাপানো

বিজ্ঞাপন কিম্বা হ্যাণ্ডবিলে, পাঠকের মনোযোগ আকর্ষণের জন্য প্রায়ই Sale, Hurry, Rush, Discount প্রভৃতি শব্দ একাধিকবার লেখা হয়। কম্পিউটারের সাহায্যে খুব সহজে এ ধরনের বিজ্ঞাপনের Art pull কিম্বা হ্যাণ্ডবিল ছাপানো যায়।

```
10 REM Program to Print handbill
20 REM A=number of handbill to be printed
30 REM B=number of times, the line "RU :H !
      ......" is to be printed
40 INPUT A
50 INPUT B
60 FOR X=1 TO A
70 PRINT TAB (10) ; "***************"
80 PRINT TAB(10) ; "* STOCK CLEARENCE
      SALE !! *"
90 PRINT TAB (10) ; "***************"
100 FOR Y=1 TO B
110 PRINT  TAB  (13) ; "RUSH ! RUSH !!
      RUSH !!!"
120 NEXT Y
130 PRINT  TAB  (6) ; "Faboulous discount on
      Electronics"
140 PRINT  TAB  (6) ; "and Computer Books.
      Attractive"
150 PRINT TAB (6) ; "FREE GIFTS with every
      purchase !"
160 PRINT TAB (6) ; "Last date : 30 March"
170 PRINT
180 PRINT TAB (10) ; "Visit FOTONSOFT Inc"
190 PRINT TAB (14) ; "Stall no : 77"
200 NEXT X
```

TAB

```
210 PRINT
220 END
RUN
? 1
? 4
    * * * * * * * * * * * * * * * * * * * * * * *
    * STOCK CLEARENCE SALE ! ! *
    * * * * * * * * * * * * * * * * * * * * * * *
         RUSH ! RUSH ! ! RUSH ! ! !
         RUSH ! RUSH ! ! RUSH ! ! !
         RUSH ! RUSH ! ! RUSH ! ! !
         RUSH ! RUSH ! ! RUSH ! ! !
      Faboulous discount on Electronics
      and Computer Books.   Attractive
      FREE GIFTS with every purchase !
      Last date : 30 March

         Visit FOTONSOFT Inc
              Stall no : 77
RUN
? 1
? 8
    * * * * * * * * * * * * * * * * * * * * * * *
    * STOCK  CLEARENCE SALE ! ! *
    * * * * * * * * * * * * * * * * * * * * * * *
         RUSH ! RUSH ! ! RUSH ! ! !
         RUSH ! RUSH ! ! RUSH ! ! !
         RUSH ! RUSH ! ! RUSH ! ! !
         RUSH ! RUSH ! ! RUSH ! ! !
         RUSH ! RUSH ! ! RUSH ! ! !
         RUSH ! RUSH ! ! RUSH ! ! !
         RUSH ! RUSH ! ! RUSH ! ! !
```

TAB

RUSH ! RUSH !! RUSH !!!
Faboulous discount on Electronics
and Computer Books. Attractive
FREE GIFTS with every purchase !
Last date : 30 March.

Visit FOTONSOFT Inc
Stall no : 77

এই প্রোগ্রামে দুটি FOR, NEXT আবর্ত ব্যবহার করা হয়েছে। প্রথমটি হল FOR X=1 TO A এবং দ্বিতীয়টি FOR Y=1 TO B. লক্ষ্য করে দেখুন, প্রথম আবর্তটির ভিতরে দ্বিতীয় আবর্তটি বসানো হয়েছে। এইভাবে আবর্ত বা Loop ব্যবহারকে বলা হয় Nested Loop Program.

Nested Loop ব্যবহার করার সময় দুটি নিয়ম সর্বদা মনে রাখতে হবে—

(1) একটি Loop-এর ভিতরে অপর একটি Loop বসানোর সময়ে খেয়াল রাখতে হবে দ্বিতীয় Loopটি যেন সম্পূর্ণভাবে প্রথম Loop-টির ভিতরে থাকে। যেমন—

```
20 FOR A=1 TO 10
30............
40............
50 FOR B=1 TO 3
60............
70 NEXT B
80............
90............
100 NEXT A
```

কিন্তু যদি কোনো প্রোগ্রামে লেখা হয়—

```
20 FOR A=1 TO 10
30
40
50 FOR B=1 TO 3
60
70 NEXT A
80
90
100 NEXT B
```

12

GOSUB

তবে দুটি Loop পরস্পর কাটাকুটি হয়ে যাওয়াতে প্রোগ্রামটি ভুল হয়ে যাবে।

(2) GOTO কিম্বা IF, THEN-এর সাহায্যে, JUMP করে FOR, NEXT Loop-এর মাঝে ঢুকে পড়া চলবে না। কম্পিউটার FOR TO বাদ দিয়ে সরাসরি NEXT স্টেটমেন্ট বোঝে না।

সেই হিসেবে নিচের প্রোগ্রামটি ভুল হবে।

```
┌──20 GO TO 80
│  30·········
│  40·········
│  50 FOR A=1 TO 10──┐
│  60·········       │
│  70·········       │
└──80·········       │
  100                │
  110 NEXT A ────────┘
```

GOSUB

অনেক সময় প্রোগ্রাম লিখতে গিয়ে দেখা যায় কোন একটি বিশেষ ফর্মুলা বা নির্দেশ, প্রোগ্রামে একাধিকবার লাগছে (ঠিক যেমন ইউটিলিটি প্রোগ্রামের ক্ষেত্রে হয়)। সাধারণ ভাবে ওই প্রোগ্রাম লিখতে গেলে, ফর্মুলাটি একাধিকবার লিখতে হয়। এতে প্রোগ্রামার খাটুনিতো বাড়েই, তাছাড়া আরো গুরুত্বপূর্ণ ব্যাপার হলো, একাধিকবার লেখা নির্দেশগুলি কম্পিউটারের অমূল্য RAM স্মৃতি ভাণ্ডার দখল করে নেয়। তাই এই ধরনের প্রোগ্রাম লেখা সহজতর করতে BASIC-এ বিশেষ Subroutine স্টেটমেন্ট, 'GOSUB'-এর ব্যবস্থা রাখা হয়েছে। GOSUB-এর সঙ্গে আর একটি স্টেটমেন্ট ব্যবহার করা হয়; সেটি হলো 'RETURN' (যেমন FOR, TOর সঙ্গে NEXT স্টেটমেন্ট ব্যবহার করা হয়)।

সাব-রুটিন বলতে বোঝায় মূল প্রোগ্রামের মধ্যে অবস্থিত একটি স্বয়ং সম্পূর্ণ ছোট প্রোগ্রাম। এই ছোট প্রোগ্রামের মধ্যে আমরা একাধিকবার প্রয়োজন হবে এমন নির্দেশ লিখে রাখি। মূল প্রোগ্রামে

১২

GOSUB

যখনই ওই নির্দেশের প্রয়োজন হয়, তখনই GOSUB স্টেটমেন্টের সাহায্যে C.P.U নিয়ন্ত্রণকে সাব-রুটিন প্রোগ্রামে পাঠিয়ে দেওয়া হয়। সাব-রুটিন প্রোগ্রামের সর্বশেষ লাইনে RETURN স্টেটমেন্ট লেখা থাকে। C.P.U সাব-রুটিন প্রোগ্রামের নির্দেশ পড়ে যখন ওই RETURN স্টেটমেন্ট আসে, তখন ওই RETURN স্টেটমেন্ট, C.P.U নিয়ন্ত্রণকে আবার মূল প্রোগ্রামে ফিরিয়ে দেয়।

এই ব্যবস্থায় সব থেকে গুরুত্বপূর্ণ সুবিধা হল C.P.U নিয়ন্ত্রণ মূল প্রোগ্রামের যে লাইন থেকে সাব-রুটিনে প্রবেশ করেছিল, RETURN-এর সাহায্যে মূল প্রোগ্রামে ফেরার সময় নিয়ন্ত্রণ ঠিক তার পরবর্তী লাইনে ফিরে আসে। ফলে সাব-রুটিন এবং মূল প্রোগ্রামের মধ্যে Loop বা আবর্তের সৃষ্টি হয় না।

মূল প্রোগ্রাম থেকে যতবার খুশি সাব-রুটিনে আসা যায়। বড় জটিল ধরনের প্রোগ্রাম লেখার সময় GOSUB স্টেটমেন্ট প্রোগ্রামারকে খুব সাহায্য করে।

GOSUB-এবং RETURN স্টেটমেন্ট লেখার নিয়ম :—

```
10 INPUT A, B
20 LET C=A+B
:
60 GOSUB 300
70
:
120 GOSUB 300
130
:
229 STOP
300 PRINT C
310 PRINT A, B
320 RETURN
```

ওপরের প্রোগ্রামে 60 স্টেটমেন্ট নাম্বারে লেখা আছে GOSUB 300। এর অর্থ হলো C.P.U নিয়ন্ত্রণ 60 নাম্বার লাইনে এলেই GOSUB 300 নির্দেশ পেয়ে সরাসরি 300 নাম্বার লাইনে চলে যাবে

(ঠিক যেমন GOTO 300 নির্দেশ পেলে যেতো)। 300 থেকে 310 লাইনের মধ্যে সাব-রুটিনের নির্দেশ লেখা হয়েছে। এই নির্দেশ পালন করার পর C.P.U নিয়ন্ত্রণ 320 লাইনে আসবে। 320-তে লেখা আছে RETURN. তাই C.P.U 320 থেকে আবার মূল প্রোগ্রামের 70 নম্বর লাইনে ফিরে যাবে (কারণ 60 নম্বর লাইন থেকে C.P.U সাবরুটিনে প্রবেশ করেছিল)। এরপর C.P.U যথারীতি মূল প্রোগ্রামের ক্রমবর্ধমান স্টেটমেন্ট নাম্বার অনুযায়ী নির্দেশ পালন করতে করতে 120 লাইনে আসবে। 120 লাইনে পুনরায় GOSUB 300 লেখা আছে। সুতরাং C.P.U নিয়ন্ত্রণ 120 লাইন থেকে সোজাসুজি সাব-রুটিনে চলে যাবে। সাব-রুটিনের নির্দেশ পালন করে 320 লাইন থেকে এবার C.P.U ফিরে আসবে 130 লাইনে। তারপর আবার মূল প্রোগ্রামের ক্রমবর্ধমান স্টেটমেন্ট নাম্বার অনুযায়ী নির্দেশ পালন করে 229 লাইনে প্রোগ্রামটি শেষ করবে।

সাব-রুটিন প্রোগ্রাম লেখার সময় তিনটি ব্যাপার খেয়াল রাখতে হবে—

(1) GOSUB-এর বদলে ভুল করে GOTO লিখে সাব-রুটিন চালু করা যাবে না। কম্পিউটার GOSUB স্টেটমেন্ট ছাড়া RETURN স্টেটমেন্টের অর্থ বুঝতে পারে না (যেমন FOR, TO না লিখলে শুধু NEXT লিখে প্রোগ্রাম চালানো যায় না)।

(2) GOTO, IF THEN ইত্যাদির সাহায্যে JUMP করে সাব-রুটিনে প্রবেশ করা চলবে না।

(3) একটি GOSUB-এর সাহায্যে আর একটি GOSUB-এ সরাসরি চলে যাওয়া যেতে পারে। ফলে ইচ্ছা করলে একাধিক পরস্পর সংযুক্ত সাব-রুটিন প্রোগ্রাম লেখা যায়।

এবার আসুন, এখনো পর্যন্ত আমাদের শেখা সমস্ত Key Word-এর ব্যবহার একটি দ্বিঘাত সমীকরণ সমাধানের সফট্ওয়্যার লিখে ঝালিয়ে নেওয়া যাক।

Quadratic Equation সমাধান করার সফট্ওয়্যার

$ax^2+bx+c=0$ ধরনের সমীকরণ গণিতের পরিভাষায় Quadratic equation (দ্বিঘাতের সহ সমীকরণ) বলা হয়। স্কুলের ছাত্রদের এই সমীকরণ সমাধান (Solve) করার অঙ্ক কষতে হয়। বস্তুত উচ্চশিক্ষায় (বিজ্ঞান বিষয়ে) প্রায়শই আমাদের Quadratic

১২

দ্বিঘাত সমীকরণ সমাধানের সফ্‌টওয়্যার

equation solve করতে হয়। সমীকরণটি 'Middle term break' পদ্ধতিতে করার মতো সহজ হলে, সমাধান মুখে মুখেই করে ফেলা যায়। তবে a, b, c ইত্যাদি co-efficient-গুলির মান বড় বা ভগ্নাংশ হলে, কাগজ কলমের সাহায্য নিতে হয়। এই সমীকরণ সমাধানের User friendly Software কিন্তু GOSUB স্টেটমেন্টের সাহায্যে খুব সহজে লেখা যায়।

Quadratic equation solve করার জন্য আমরা শ্রীধর আচার্যের ফর্মুলা ব্যবহার করে থাকি। যাঁরা ছোটবেলায় কষা এই অঙ্ক ভুলে গেছেন, তাঁদের মনে করিয়ে দিই—

ধরা যাক $ax^2+bx+c=0$ একটি সমীকরণ

এই সমীকরণ থেকে আমাদের x-এর মান a, b, c-র সাহায্যে নির্ণয় করতে হবে। সমীকরণটি যেহেতু একটি 2nd order equation, তাই এর থেকে x-এর সর্বাধিক দুটি মান পাওয়া যাবে।

শ্রীধর আচার্যের ফর্মুলা অনুযায়ী x-এর সেই দুটি মান হবে—

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

অর্থাৎ $$x=\frac{-b+\sqrt{b^2-4ac}}{2a}$$

এবং $$x=\frac{-b-\sqrt{b^2-4ac}}{2a}$$

আসল অঙ্কে a, b, c-র মান সংখ্যায় দেওয়া থাকবে। সেই মানগুলি ফর্মুলায় বসালেই x-এর দুটি মান বেরিয়ে যাবে।

এই ফর্মুলাটি আমাদের প্রোগ্রামে একটি সাব-রুটিনে লেখা হবে, কারণ প্রোগ্রামের মূল অংশে দুটি ভাগ থাকবে। একটি ভাগ x-এর Real মান বার করতে কাজ করবে এবং অপর ভাগ x-এর Imaginary মান বার করতে কাজ করবে। যে অঙ্কে (b^2-4ac) অংশটির মান Positive হবে, সেই অঙ্কে x-এর মান Real হবে। আর যে অঙ্কে (b^2-4ac)-র মান negative, সেই অঙ্কে x-এর মান Imaginary হবে।

Quadratic equation solve করার Software-টি তিনটি সাব-রুটিনে আলাদা করা হয়েছে। প্রথম সাব-রুটিনে আছে শ্রীধর আচার্যের ফর্মুলা। তবে এই ফর্মুলাটি একটু বদলে লিখতে হয়েছে।

52

দ্বিঘাত সমীকরণ
সমাধানের সফট্ওয়্যার

মূল ফর্মূলায় আছে $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

আর আমি লিখেছি $x = \frac{-b \pm \sqrt{|b^2 - 4ac|}}{2a}$

যেখানে $|b^2 - 4ac|$ বলতে বোঝানো হয়েছে, $b^2 - 4ac$ যদি negetive-ও হয়, তাহলেও $|b^2 - 4ac|$-র মান Positive হবে অর্থাৎ অঙ্কের ভাষায় যাকে Modulus বলে।

এইভাবে লেখার কারণ $(b^2 - 4ac)$ যদি কোন অঙ্কে ঋণাত্মক (negative) হয় তবে, $\sqrt{b^2 - 4ac}$-টি imaginary হয়ে যবে। কম্পিউটার কোনো imaginary সংখ্যার বর্গমূল নির্ণয় করতে পারে না। তাই X-এর মান imaginary হলে কম্পিউটার শ্রীধর আচার্যের ফর্মূলা সরাসরি ব্যবহার করে ওই অঙ্ক কষতে পারবে না।

মূল প্রোগ্রামে IF, THEN Branching-এর সাহায্যে $(b^2 - 4ac)$র মান positive না negative সেটি পরীক্ষা করা হচ্ছে এবং এই ফলাফল অনুযায়ী C.P.U নিয়ন্ত্রণকে অন্য দুটি সাব-রুটিনে পাঠানো হচ্ছে। যদি $(b^2 - 4ac)$ negative হয়, তবে নিয়ন্ত্রণ যাচ্ছে দ্বিতীয় সাব-রুটিনে, এবং যদি $(b^2 - 4ac)$ positive হয়, তবে যাচ্ছে তৃতীয় সাব-রুটিনে। এই দুটি সাব-রুটিন, X-এর মান Real না Imaginary সেই অনুসারে উত্তর ছাপানোর আদেশ দিচ্ছে।

একটি অঙ্ক কষার পর প্রোগ্রামার আর একটি করবেন কিনা, সেই আদেশ গ্রহণ করছে মূল প্রোগ্রাম এবং প্রোগ্রামারের নির্দেশানুসারে পুরো প্রোগ্রামটি পুনরায় গোড়া থেকে শুরু হচ্ছে, কিম্বা শেষ হয়ে যাচ্ছে।

```
10 REM Program to Solve Quadratic Equation.
20 PRINT TAB (5) ; "Let us solve a Quadratic
        Equation."
30 PRINT TAB (5) ; "Written in the form :—"
40 PRINT TAB (15) ; "AX∧2+BX+C=0"
50 PRINT "Type the values of A, B, C after the
        Question mark"
```

১২

দ্বিঘাত সমীকরণ
সমাধানের সফ্‌ট্‌ওয়্যার

```
60 INPUT A, B, C
70 GOSUB 310
80 REM Cheking whether X is Real or Complex
90 IF G> =0 THEN 120
100 GOSUB 410
110 GOTO 140
120 GOSUB 500
130 GOTO 140
140 PRINT
150 PRINT "Want to calculate more ? YES or
         NO" ;
160 INPUT A $
170 IF A $="YES" THEN 50
180 PRINT TAB (15) ; "Good bye"
200 STOP
300 REM FIRST SUBROUTINE STARTS
         HERE
310 LET G=B*B-(4*A*C)
320 LET D=ABS (G)
330 LET Y= -B/2*A
340 LET Z=SQR (D)/(2*A)
350 RETURN
400 REM SECOND SUBROUTINE STARTS
         HERE
410 PRINT TAB (10) ; "Roots are Complex"
420 PRINT TAB (5) ; "X1=" ; Y ; "+i" ; Z
430 PRINT TAB (5) "X2=" ; Y ; "-i" ; Z
440 RETURN
500 REM THIRD SUBROUTINE STARTS
         HERE
510 PRINT TAB (10) ; "Roots are Real"
520 PRINT TAB (10) ; "X1=" ; Y+Z
```

যে কোনো negative অর্থাৎ ঋণাত্মক সংখ্যার বর্গমূল (Square Root), অঙ্কের পরিভাষায় Imaginary কিম্বা Complex বলা হয়। সেই হিসেবে $\sqrt{-5}$ একটি Imaginary অথবা Complex সংখ্যা। অঙ্ক কষার সময়ে এই সংখ্যাগুলিকে সহজে বোঝার জন্য এই ভাবে লেখা হয়—

$$\sqrt{-5} = \sqrt{-1} \times \sqrt{5} = i\sqrt{5}$$

যেখানে $\sqrt{-1} = i$ (অনেক সময় $\sqrt{-1}$-কে i-এর পরিবর্তে j হরফ দিয়েও বোঝানো হয়)।

```
530 PRINT TAB (10) ; "X2 = " ; Y—Z
540 RETURN
550 END
END
```

Let us solve a Quadratic Equation
written in the form :—

AX∧2+BX+C=0

Type the values of A, B, C after the question mark

? 1, 4, 0

Roots are Real

X1=0

X2=−4

Want to calculate more ? YES or NO ? YES

Type the values of A, B, C after the question mark

? 9, 4, 1

Roots are Complex

×1=−0·222222222+i 0.248451998

×2=−0.222222222−i 0.248451998

Want to calculate more ? YES or NO ? NO

Good Bye

১২

READ, DATA

READ

DATA

আমরা ইতিপূর্বে দেখেছি কি ভাবে INPUT স্টেটমেন্টের সাহায্যে কম্পিউটারকে অজানা variable-এর মান জানতে চাওয়ার নির্দেশ দেওয়া হয়। INPUT নির্দেশ পেলে, কম্পিউটার প্রতিবার প্রোগ্রামটি

১২

READ, DATA

RUN করা হলে প্রোগ্রামার কাছে একটি করে ভেরিয়েবল-এর মান জানতে চায়। প্রোগ্রামার ওই মানটি কী বোর্ডে টাইপ করলে কম্পিউটার হিসেব করে ফলাফল বলে দেয়। যেমন—

```
10 INPUT A
20 PRINT A*2
30 END
RUN
? 2
4
RUN
? 3
6
```

এই ব্যবস্থা সাধারণ কাজের পক্ষে খুবই উপযোগী। কিন্তু যখন একই প্রোগ্রামের সাহায্যে অনেকগুলি হিসেব করতে হয়, তখন INPUT স্টেটমেন্টের মাধ্যমে কম্পিউটারকে তথ্য জোগান দিতে অনেক সময় লাগে। কারণ প্রতিটি তথ্য টাইপ করে, কম্পিউটারের উত্তর বলে দেওয়ার জন্য প্রোগ্রামারকে বসে থাকতে হয়। ফলাফল প্রকাশের পর প্রোগ্রামারকে নতুন করে RUN কম্যাণ্ড দিয়ে তথ্য টাইপ করতে হয়। অথচ যদি প্রোগ্রামের মধ্যেই একসঙ্গে সমস্ত INPUT তথ্য লিখে দেওয়া যেতো, আর কম্পিউটার স্বয়ংক্রিয়ভাবে ওই তালিকা থেকে পর পর একটি করে variable-এর মান পড়ে নিয়ে ফলাফল জানিয়ে দিতো, তবে প্রোগ্রামারের সময় অনেক বাঁচতো, (ঠিক যেন বাধ্য ছাত্রকে Home task দিয়ে বাড়ি পাঠিয়ে দেওয়া, সামনে বসে খবরদারী করার আর দরকার নেই।) BASIC ভাষা তৈরী করার সময় এই কথা চিন্তা করেই READ, DATA স্টেটমেন্ট দুটির ব্যবস্থা রাখা হয়েছে।

GOSUB, RETURN-এর মতোই READ, DATA একত্রে ব্যবহার করা হয়। যে সব প্রোগ্রামে READ, DATA স্টেটমেন্টের প্রয়োজন, সেইসব প্রোগ্রামে INPUT স্টেটমেন্টের বদলে READ স্টেটমেন্টটি বসাতে হয়। যেমন, আগের প্রোগ্রামটিতে যদি READ, DATA ব্যবহার করতে হয় তবে

52

READ, DATA

10 নম্বর লাইনে লিখতে হবে—

```
10 READ A
```

এই নির্দেশ পেলেই কম্পিউটার বুঝে যাবে variable A-র মান প্রোগ্রামের ভিতরই DATA লেখা লাইনে লেখা আছে। সুতরাং কম্পিউটার 10 নম্বর লাইন RUN করার পর V.D.U স্ক্রিনে '?' চিহ্ন না ফুটিতে তুলে প্রোগ্রামের মধ্যে DATA লেখা লাইনটি খুঁজে বার করবে, এবং সেখান থেকে তথ্য পড়ে পর পর ফলাফল ছাপিয়ে দেবে। DATA স্টেটমেন্টটি এইভাবে লিখতে হবে—

```
25 DATA 2, 3
```

এর অর্থ হলো, A variable-এর প্রথম মান 2 এবং দ্বিতীয় মান 3

তাহলে প্রোগ্রামটি হবে এই রকম—

```
10 READ A
20 PRINT A*2
25 DATA 2, 3
30 END
```

তবে READ, DATA স্টেটমেন্ট এইভাবে সরাসরি কাজ করে না। উপরের প্রোগ্রামটি লিখে RUN করলে দেখা যাবে কম্পিউটার কেবলমাত্র A=2 মানটি পড়ে উত্তর বলে দিয়েছে, আর DATA লাইনে A=3 যে তথ্যটি লেখা আছে, সেটিকে একদম অগ্রাহ্য করেছে। এর কারণ শুধুমাত্র READ, DATA নির্দেশ পেলে কম্পিউটার DATA লাইন থেকে একবার তথ্য পড়বে। সবকটি তথ্য কম্পিউটারকে পড়াতে গেলে READ স্টেটমেন্টটিকে একটি FOR, TO, NEXT আবর্তের মধ্যে বসাতে হবে।

অর্থাৎ প্রোগ্রামটি লিখতে হবে এইভাবে :—

```
5 FOR X=1 TO 2
10 READ A
15 PRINT A*2
20 NEXT X
```

```
25 DATA 2, 3
30 END
```

এবার প্রোগ্রামটি RUN করলে দেখা যাবে কম্পিউটার দুটি তথ্যই পড়ে উত্তর ছাপিয়ে দিয়েছে।

```
RUN
 4
 6
```

আর একটি READ, DATA-র উদাহরণ দেখা যাক—

```
NEW
 5 DATA 4, 5, 7, 8, 3, 2, 0, 5
10 FOR B=1 TO 8
20 READ A
30 PRINT A
40 NEXT B
50 END
RUN
 4
 5
 7
 8
 3
 2
 0
 5
```

এই প্রোগ্রামে একটি বৈশিষ্ট্য নিশ্চয়ই লক্ষ্য করেছেন, সেটি হলো প্রোগ্রামের শুরুতেই DATA স্টেটমেন্টটি লেখা হয়েছে। বস্তুত DATA স্টেটমেন্টটি প্রোগ্রামের যে কোন জায়গায় লেখা যায়।

আর একটি ব্যাপার লক্ষ্য করুন। DATA লাইনে মোট আটটি তথ্য দেওয়া আছে বলে FOR, NEXT আবর্তটি 8 বার ঘোরানোর

READ DATA

নির্দেশ দেওয়া হয়েছে। যদি FOR, NEXT আবর্তটি আরো কম বার ঘোরানো হোত তবে কম্পিউটার কম তথ্য ছাপতো। যেমন—

```
 5 DATA 4, 5, 7, 8, 3, 2, 0, 5
10 FOR B=1 TO 4
20 READ A
30 PRINT A
40 NEXT B
50 END
RUN
 4
 5
 7
 8
```

READ, DATA-র সাহায্যে, একটি সংখ্যার তালিকা থেকে বৃহত্তম সংখ্যাটি খুঁজে বার করা।

অনেকগুলি সংখ্যার একটি তালিকা থেকে সর্ববৃহৎ সংখ্যাটি কাগজ কলমের সাহায্য নিয়ে নির্ণয় করতে বেশ সময় লাগে। তাছাড়া এতে ভুল হবার সম্ভবনাও থাকে প্রচুর। অথচ এই কাজটি কম্পিউটারের সাহায্যে খুব সহজে আর অত্যন্ত দ্রুত করে ফেলা যায়।

```
NEW
10 DATA 101, 203, 4, 121, 230, 229, 179, 65,
   231, 5, 64, -1
20 READ A
30 READ B
40 REM Condition to Check the end of data
50 IF B<O THEN 100
60 IF B>A THEN 80
70 GOTO 30
80 LET A=B
90 GOTO 30
100 PRINT A
```

```
110 END
RUN
231
```

এই প্রোগ্রামে দুটি বৈশিষ্ট্য লক্ষ্য করার মতো। প্রথমত, প্রোগ্রামটিতে FOR, NEXT loop ব্যবহার করা হয়নি। তার পরিবর্তে 70 এবং 90 লাইনে GOTO স্টেটমেন্টের সাহায্যে প্রোগ্রামটিকে আবর্তে ঘোরানো হয়েছে। দ্বিতীয়ত, এই প্রোগ্রামের DATA লাইনে যত খুশি সংখ্যা লেখা যাবে। তার জন্য কম্পিউটারকে কতবার আবর্তে ঘুরতে হবে, সেটি জানাতে হবে না। এটি সম্ভব হয়েছে DATA লাইনে সংখ্যার তালিকার শেষে লেখা −1 সংখ্যাটি এবং 50 লাইনে লেখা IF, THEN স্টেটমেন্টের ব্যবহারের ফলে।

−1 সংখ্যাটিকে, কম্পিউটার পরিভাষায় যাকে বলে dummy data, সেইভাবে ব্যবহার করা হয়েছে। লক্ষ্য করে দেখুন, এই প্রোগ্রামের DATA লাইনে যতগুলি সংখ্যা লেখা হয়েছে, তারা সবই Positive বা ধনাত্মক সংখ্যা। কিন্তু সর্বশেষ সংখ্যাটির শেষে ইচ্ছে করে একটি অপ্রয়োজনীয় negative বা ঋণাত্মক সংখ্যা লেখা হয়েছে। এছাড়া 50 নাম্বার স্টেটমেন্টের মাধ্যমে কম্পিউটারকে নির্দেশ দেওয়া হয়েছে, 'তালিকায় negative সংখ্যা দেখলেই ফলাফল ছেপে দাও'। ফলে কম্পিউটার তালিকা থেকে যতক্ষণ না সর্বশেষ negative সংখ্যাটি পড়ছে, ততক্ষণ স্বয়ংক্রিয়ভাবে READ, DATA-র আবর্ত চালু থাকছে।

DIM

বড় বড় এক স্তম্ভ (Column) বিশিষ্ট তালিকা READ, DATA স্টেটমেন্টের সাহায্যে তৈরী করা খুব সহজ। তবে একাধিক স্তম্ভ বিশিষ্ট তালিকা ছাপাতে হলে শুধু READ, DATA স্টেটমেন্ট ব্যবহার করলে কাজ হয় না। এর জন্য ব্যবহার করতে হয় DIM স্টেটমেন্ট। ইংরিজি শব্দ Dimension-কে সংক্ষিপ্ত করে BASIC-এর DIM স্টেটমেন্টের নামকরণ করা হয়েছে।

যাঁরা B.Sc. পর্যন্ত অঙ্ক কষেছেন, তাঁদের DIM স্টেটমেন্টের ব্যবহার বুঝতে খুব একটা সময় লাগবে না, কারণ তাঁরা ইতিমধ্যেই

Determinant কিম্বা Matrix-এর অঙ্ক কষে গাণিতিক Array-র সঙ্গে পরিচিত। অবশ্য এমন অনেক পাঠক আছেন, যাঁরা Array বলতে কি বোঝায় তা জানেন না, তাঁদের জন্য নিচে সংক্ষেপে Array সম্বন্ধে বলা হলো।



DIM

ধরা যাক আপনাকে স্কুলের পরীক্ষার ফলাফল Tabulate করতে বলা হয়েছে। সমস্ত পরীক্ষক তাঁদের দেখা পরীক্ষার খাতা আপনাকে দিয়ে দিয়েছেন। এবার তাহলে আপনি কি করবেন? একটা সাদা কাগজে দাগ কেটে ক্লাস, বিভাগ, ছাত্রের নাম, রোল নাম্বার এবং পরীক্ষালব্ধ ফলের জন্য পাঁচটি Column বা স্তম্ভ বানাবেন। তারপর ওই সমস্ত স্তম্ভে যথাযথ তথ্য বসিয়ে Tabulation-এর কাজ শেষ করবেন, তাই না?

অঙ্কের ভাষায় এইভাবে নির্দিষ্ট ছকে সাজানো তথ্যের তালিকা বা সারণীকে বলা হয় Array। যে কোন Array-তেই একটি বৈশিষ্ট্য লক্ষ্য করার মতো, সেটি হলো, স্তম্ভে লিখিত তথ্যগুলি সবই একই ধরনের তথ্য। যেমন বয়সের স্তম্ভে কেবল বয়সই লেখা থাকে, নামের স্তম্ভে নাম। তাই Array-র স্তম্ভে লেখা তথ্যগুলি একই নিউমেরিক বা স্ট্রিং ভেরিয়েবল দিয়ে চিহ্নিত করা হয়। স্তম্ভের মধ্যে তথ্যের অবস্থান নির্দেশ করার জন্য ভেরিয়েবলগুলির সঙ্গে একটি Suffix বা Subscript লেখা হয়। সেই জন্য অনেক সময় Array-র ভেরিয়েবলগুলিকে Subscripted Variableও বলা হয়। BASIC ভাষায় সমস্ত Subscripted Variable লেখা হয় এইভাবে :—

A(1), A(2), A(3) ইত্যাদি

কম্পিউটারের মাধ্যমে কোন Array সৃষ্টি করতে গেলে DIM স্টেটমেন্টের মাধ্যমে কম্পিউটারকে জানিয়ে দিতে হয় Arrayটিতে কতগুলি Subscripted Variable আছে। কম্পিউটার সেই তথ্য অনুসারে আভ্যন্তরীণ স্মৃতিতে ততোগুলি মেমরী সেল ওই Subscripted Variable লেখার জন্য বরাদ্দ করে দেয়। এতে প্রোগ্রামারের কাজের খুব সুবিধা হয়। যেমন ধরুন একটি 100 ভেরিয়েবলের Array থেকে 95-তম ভেরিয়েবলের মান জানতে গেলে কম্পিউটারে কেবলমাত্র 95-তম Subscripted Variable-কে

১২

DIM

PRINT করার নির্দেশ দিলেই চলে। কিন্তু যদি DIM স্টেটমেন্ট ব্যবহার করা না হয়, তবে কম্পিউটার Array থেকে কোন বিশেষ তথ্য খুঁজে বার করতে পারে না।

নিচে DIM স্টেটমেন্ট ব্যবহারের ছুটি উদাহরণ দেওয়া হলো।

```
NEW
10 REM  Demonstration program of DIM State-
        ment
20 DIM A(10)
30 FOR B=1 TO 10
40 READ A(B)
50 PRINT A(B)
60 NEXT B
80 DATA 1, 6, 3, 9, 4, 2, 7, 6, 5, 8
90 END
RUN
 1
 6
 3
 9
 4
 2
 7
 6
 5
 8
```

এই প্রোগ্রামে 10টি Subscripted Variable-এর Array গঠন করা হয়েছে। এবার দেখা যাক, কিভাবে 1, 5, 7, 10 ইত্যাদি Subscripted Variableগুলির মান আমরা কম্পিউটারের মাধ্যমে ছাপতে পারি।

52
DIM

কী বোর্ডে টাইপ·করুন

```
50
61 PRINT A (1)
62 PRINT A (5)
63 PRINT A (7)
64 PRINT A (10)
```

সুতরাং পরিবর্তিত প্রোগ্রামটি কি দাঁড়ালো, সেটি LIST-এর সাহায্যে দেখে নিন্—

```
LIST
10 REM  Demonstration  program  of DIM State-
        ment
20 DIM A(10)
30 FOR B=1 TO 10
40 READ A (B)
60 NEXT B
61 PRINT A (1)
62 PRINT A (5)
63 PRINT A (7)
64 PRINT A (10)
80 DATA 1, 6, 3, 9, 4, 2, 7, 6, 5, 8
90 END
RUN
 1
 4
 7
 8
```

DIM স্টেটমেন্টের সাহায্যে আমরা একটি Arrayর মান অন্য Arrayতে assign করতে পারি। যেমন এই A (1) থেকে A (10) Arrayর সম মানের আর একটি Array যদি আমরা তৈরী করতে চাই তবে প্রোগ্রামটি হবে এই রকম—

১২

DIM

কী বোর্ডে টাইপ করুন —

```
10 DIM A (10), C (10)
20
61 FOR D=1 TO 10
62 LET C (D)= A (D)
63 PRINT A (D) ; C (D)
64 NEXT D
LIST
10 DIM A (10), C (10)
30 FOR B=1 TO 10
40 READ A (B)
60 NEXT B
61 FOR D=1 TO 10
62 LET C (D)= A (D)
63 PRINT A (D) ; C (D)
64 NEXT D
80 DATA 1, 6, 3, 9, 4, 2, 7, 6, 5, 8
90 END
RUN
 1      1
 6      6
 3      3
 9      9
 4      4
 2      2
 7      7
 6      6
 5      5
 8      8
```

১২

Debugging

DEBUGGING

নোতুন প্রোগ্রাম লিখে কী বোর্ডে টাইপ করা মাত্রই সব সময় সেটি ঠিক মতো কাজ করে না আর তখনই দরকার হয় Debugging. Debugging কথাটির আক্ষরিক বাংলা অনুবাদ করলে দাঁড়ায় 'উকুন বাছা' ধরনের ব্যাপার! কম্পিউটার পরিভাষায় এর অর্থ হলো প্রোগ্রামের ভুল ত্রুটি খুঁজে বার করা। তবে কম্পিউটার প্রোগ্রাম থেকে ভুল বার করা উকুন বাছার থেকেও ঢের বেশী কঠিন কাজ।

মানুষের ভাষায় কেবল দুটি ভুল হতে পারে, (1) বানান ভুল আর (2) ব্যাকরণগত ত্রুটি। কিন্তু কম্পিউটার প্রোগ্রামে আরো একটি ভুলের উৎস থাকে, যেটি খুঁজে বার করতে প্রোগ্রামারদের মাঝে মধ্যেই কালঘাম ছুটে যায়! এটি হলো Logical error.

যে কোনো প্রেগ্রাম লেখার সময় আমরা মনে মনে একটি যুক্তি-গ্রাহ্য পথ বা Logical path তৈরী করে নিই। অর্থাৎ আগে থেকে ভেবে নিই, "আমি যদি এই নির্দেশ দিই, তবে কম্পিউটার ঠিক কি কাজ করবে।" এই ভাবনার মধ্যেই যদি কোনো ভুল থেকে যায়, তবে ভুল প্রোগ্রামটি বার বার পড়েও সেই ভুলটি ধরতে পারা যায় না।

অপর পক্ষে প্রোগ্রাম লেখায় যদি ব্যাকরণগত ত্রুটি থাকে (যেমন কী ওয়ার্ড লেখার সময় বানান ভুল), যাকে কম্পিউটার পরিভাষায় বলা হয় Syntax, খুব সহজেই এক নজরে ধরা পড়ে যায়। তাছাড়া Syntax error খুঁজে বার করার জন্য সমস্ত আধুনিক মাইক্রো কম্পিউটারের OS বিশেষ Debugging প্রোগ্রাম ভরা থাকে। এই প্রোগ্রাম, প্রোগ্রামারের লেখা সফট্ওয়্যারের Syntax ত্রুটি খুঁজে বার করে আর সেই সঙ্গে স্ক্রিনে ফুটিয়ে তোলে কোন লাইনে ভুলটি হয়েছে!

ধরুন আপনি কোনো প্রোগ্রাম লেখার সময় Syntax error করলেন (যেমন INPUT লেখার পরিবর্তে লিখলেন IMPUT অথবা PRINT A, B লেখার জায়গায় লিখলেন PRINT A : B)। এই ভুলযুক্ত প্রোগ্রামটি যে কোনো কম্পিউটারে RUN করলেই কম্পিউটার মনিটর স্ক্রিনের মাধ্যমে আপনাকে জানিয়ে দেবে

কম্পিউটার পরিভাষায় Debugging শব্দটির সংযোজনের ঘটনা রীতিমত চমকপ্রদ। যে Mark I কম্পিউটারের কথা আমরা আগেই জেনেছি, সেই কম্পিউটারটি একদিন চলতে চলতে হঠাৎ বন্ধ হয়ে গেল। ঘটনাটি 1945 সালে ঘটেছিল। যন্ত্রপাতি ঘাঁটাঘাঁটির পর আবিষ্কার হল একটি মথ্ (Moth) পোকা কম্পিউটারটির বড় রিলের কনট্যাক্টের ফাঁকে ঢুকে রিলের কনট্যাক্টের চড় চাপড় খেয়ে চেপ্টে মরে গেছে। এইজন্যই কম্পিউটারটি বিকল হয়ে পড়েছে।

ক্যাপ্টেন গ্রেস মারী হপার সেই ঐতিহাসিক মথটিকে সেদিনের লেজার খাতার পাতায় সেঁটে দিয়েছিলেন এবং উর্ধতন অফিসার এলে তাঁকে রিপোর্ট দিয়েছিলেন আজকে কম্পিউটারকে debugging অর্থাৎ পোকা মুক্ত করা হলো। সেই থেকেই debugging কথাটির প্রচলন শুরু হল।

১২

Debugging

Syntax Error হয়েছে এবং সেই সঙ্গে কোন লাইনের কোন জায়গায় ভুল হয়েছে তাও লিখে দেবে। তবে logical error খুঁজে বার করার ক্ষমতা কম্পিউটারের নেই। তাই Syntax-এর দিক থেকে নির্ভুল অথচ logic-এর দিক থেকে ত্রুটিযুক্ত প্রোগ্রাম সব সময় কম্পিউটারে RUN করা যায় কিন্তু প্রোগ্রামটি ঠিক মতো কাজ করে না। এই অসুবিধার দরুন প্রোগ্রামারের পক্ষে logical error খুঁজে বার করা আরও কঠিন হয়ে পড়ে। নিচে একটি Syntax error-যুক্ত প্রোগ্রামকে কি ভাবে Debug করা হলো তা দেখানো হলো।

```
    10LET A=3
    20LED B=6
    30PRINT "A+B="; A+B
    40PRINT "A-B="; A+B
    50PRINT "A*B="; A*B
    60PRINT "A/B="; A/B
    70END
>VDU2
>RUN

Mistake at line 20
>  20LET  B=6
>RUN
A+B=9
A-B=9
A*B=18
A/B=0.5
>  40PRINT "A-B="; A-B
>RUN
A+B=9
A-B=-3
A*B=18
A/B=0.5
```

প্রোগ্রামের 20 নম্বর লাইনে Syntax error ছিলো। ওই লাইনে LET স্টেটমেন্টটির বানান ভুল করে LED টাইপ করা হয়েছিল। কম্পিউটারে প্রোগ্রামটি RUN করার সাথে সাথে তাই মনিটর স্ক্রিনে "Mistake at line 20" কথাটি ফুটে উঠলো।

প্রোগ্রামার তখন LED বানানটি সংশোধন করে প্রোগ্রামটি RUN করলেন, দেখা গেল কম্পিউটার প্রোগ্রামটি RUN করছে। অর্থাৎ প্রোগ্রামটি Syntax-এর দিক থেকে নির্ভুল হয়েছে, সেটি বোঝা

এ পাতার প্রোগ্রামটি B.B.C Unicorn কম্পিউটারে RUN করা হয়েছে। B.B.C কম্পিউটারে Syntax error জানান দেওয়ার জন্য প্রিন্টারে ছাপা হয়েছে Mistake at line 20. কিন্তু এই প্রোগ্রামটি অন্য কোম্পানির কম্পিউটারে RUN করলে এই একই Message Report পাওয়া যাবে না। Message-এর কথাগুলি কোম্পানি ও মডেল অনুসারে পরিবর্তিত হবে।

প্রোগ্রাম ও RUN-এর মাঝে VDU 2 লেখাটি B.B.C-র একটি বিশেষ OS কম্যাণ্ড। এর সঙ্গে BASIC প্রোগ্রামটির কোনো সম্পর্ক নেই। প্রোগ্রামটি অন্য কম্পিউটারে চালানোর সময় VDU 2-কে সম্পূর্ণ অগ্রাহ্য করতে হবে।

১২

Debugging

গেল। কিন্তু বিভিন্ন তথ্য যোগান দিয়ে দেখা গেল উত্তরে ভুল হচ্ছে।

যেমন A=3, B=6 হওয়া সত্বেও কম্পিউটার ছাপছে

A+B=9

A−B=9

সুতরাং এবারকার ভুলটি হলো logical error. দেখা যাচ্ছে A+B উত্তরটি সঠিক হচ্ছে কিন্তু ভুল হচ্ছে A−B-র মানটি। সুতরাং প্রোগ্রামার বুঝলেন A−B নির্ণয় করার নির্দেশেই Logical error অথবা 'Bug'-টি লুকিয়ে আছে। তিনি তখন প্রোগ্রামটির 40 নম্বর লাইনটি আবার করে পড়ে দেখলেন (কারণ ওই লাইনেই A−B-র মান ছাপাবার নির্দেশ দেওয়া হয়েছে) দেখা গেলো, 40 নম্বর লাইনে লেখা হয়েছে

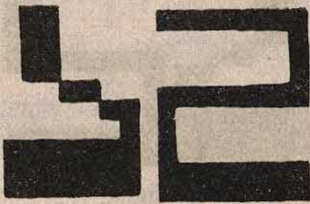
```
40 PRINT "A−B="; A+B
```

লক্ষ্য করে দেখুন, নির্দেশটি গাণিতিক যুক্তির দিক থেকে সম্পূর্ণ ভুল অথচ BASIC প্রোগ্রাম লেখার নিয়ম অনুসারে সম্পূর্ণ নিখুঁত অর্থাৎ Syntax-এর দিক থেকে নির্ভুল। সুতরাং প্রোগ্রামার এই যুক্তিগত ত্রুটিটি সংশোধন করলেন

40 PRINT "A−B="; A−B টাইপ করে।

এবার সংশোধিত প্রোগ্রামটি ফের RUN করা হলো। দেখা গেল উত্তরটি সঠিক হচ্ছে।

এটি একটি অত্যন্ত সহজ উদাহরণ। বাস্তব ক্ষেত্রে প্রোগ্রামের মধ্যে এমন জটিল সব logical error সৃষ্টি হয় যা খুঁজে বার করতে অভিজ্ঞ প্রোগ্রামারের বহু দিন লেগে যায়। অবিশ্বাস্য হলেও একটি ভিডিও গেমের প্রোগ্রাম তৈরী করে তাকে debugging করতে বিখ্যাত সফট্‌ওয়্যার ডিজাইনারদের কম করে তিন চার মাস সময় লেগে যায়! Debugging-এর ক্ষমতা ক্রমশই অভিজ্ঞতার সঙ্গে সঙ্গে বাড়ে। এ বিষয়ে সাফল্য লাভের সহজতম উপায় হলো নিজেকে কম্পিউটার হিসেবে চিন্তা করা। ধরুন একটি প্রোগ্রাম আপনাকে debug করতে হবে। আপনাকে তখন সেই প্রেগ্রামের প্রথম লাইন থেকে পড়তে শুরু করতে হবে এবং সেই সঙ্গে প্রতি লাইনে লেখা নির্দেশ মনে মনে পালন করার চেষ্টা



Debugging

করতে হবে ঠিক যেভাবে কম্পিউটারের C.P.U. ওই নির্দেশ পালন করবে। এই পদ্ধতি অনুসরণ করে ঠাণ্ডা মাথায় প্রথম থেকে শেষ লাইন অবধি নির্দেশ যদি পড়ে ফেলতে পারেন, তবে বেশির ভাগ সময়ই logical error খুঁজে বার করে ফেলা সম্ভব হয়। অবশ্য বড় প্রোগ্রামের ক্ষেত্রে এই কায়দাটি বিশেষ কাজে লাগে না। সে ক্ষেত্রে সম্পূর্ণ প্রোগ্রামটিকে অনেকগুলি ছোটো অংশে ভাগ করে নিয়ে প্রতিটি অংশ আলাদা ভাবে কম্পিউটারে RUN করে দেখতে হয় প্রতিটি অংশে সঠিক ফলাফল পাওয়া যাচ্ছে কিনা।

প্রত্যেক প্রোগ্রামারের debugging করার নিজস্ব কায়দা থাকে। আপনিও যখন debugging করবেন, তখন দেখবেন আপনি আপনার নিজস্ব কায়দায় debugging করছেন। সর্বদা মনে রাখবেন debugging প্রায়শই খুব সময় সাপেক্ষ এবং মাথা গণ্ডোগোল করে দেওয়ার মতো ব্যাপার হয়, আর খুব কম সংখ্যক বড়ো প্রোগ্রামই debugging ছাড়া প্রথম প্রচেষ্টাতেই RUN করা যায়। তাই আপনার নিজের লেখা প্রথম প্রোগ্রামটি প্রথম প্রচেষ্টাতে সফল না হলে দমে যাবেন না। দ্বিগুণ উৎসাহ নিয়ে ঠাণ্ডা মাথায় debugging করতে শুরু করে দেবেন!

পরিচ্ছেদ ১৩

প্রোগ্রামিং অ্যানালিসিস এবং ফ্লোচার্ট

ইতিপূর্বে যে সমস্ত সফট্‌ওয়্যার নিয়ে আলোচনা হয়েছে, সেগুলি নেহাতই ছোট এবং সহজ। এইসব সহজ সফট্‌ওয়্যার লেখার সময় আমরা পুরো ব্যাপারটি এক সঙ্গে মনে মনে ভেবে নিয়েছি কার পরে কি করতে হবে এবং সেই অনুযায়ী সরাসরি BASIC ভাষায় প্রোগ্রামটি লিখে ফেলেছি। কিন্তু বেশীর ভাগ বড় সফট্‌ওয়্যারে ব্রাঞ্চিং (Branching) এবং লুপিং (Looping) এতো বেশী হয় যে সব কিছু এক সঙ্গে মনে রাখা অসম্ভব হয়ে পড়ে। তাই এ ধরনের বড় সফট্‌ওয়্যার লেখার সময় আমাদের পর পর তিনটি কাজ করতে হয়, (1) প্রোগ্রামিং অ্যানালিসিস (Programing analysis), (2) প্রোগ্রাম প্রিপারেশন (Program preparation) এবং (3) ডিবাগিং ও মেনটেনেন্স (Debugging and Maintainance)। মজার কথা কি জানেন যে কোনো কম্পিউটার সফট্‌ওয়্যার লেখার সময়ই কিন্তু আমাদের এই তিনটি ধাপ পেরিয়ে আসতে হয়। ছোট প্রোগ্রামের ক্ষেত্রে আমাদের মস্তিষ্ক ওই তিনটি কাজ আমাদের অজান্তে একই সঙ্গে করে ফেলে, ফলে আমরা বুঝতেও পারি না এতো সব কাণ্ডকারখানা আমাদের মাথায় মুহূর্তের মধ্যে ঘটে গেল। তবে বড় প্রোগ্রামের

ক্ষেত্রে এই তিনটি কাজ আমাদের আলাদাভাবে ক্রমান্বয়ে করে নিতে হয়। এবার দেখা যাক ওই তিনটি ধাপে আমাদের কি কি করতে হয়।

প্রোগ্রামিং অ্যানালিসিস

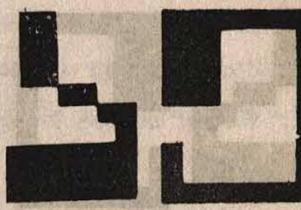
কোনো অঙ্ক কষার আগে আমাদের যেমন সর্বপ্রথম প্রশ্নপত্রটি অত্যন্ত মনোযোগ সহকারে পড়ে বুঝে নিতে হয় প্রশ্নকর্তা ঠিক কোন্ সমস্যার সমাধান চাইছেন, ঠিক তেমনি সফট্ওয়্যার রচনার আগে প্রোগামারকে বুঝে নিতে হয় কোন্ সমস্যার সমাধানের সফট্ওয়্যার তাঁকে লিখতে হবে। কম্পিউটার পরিভাষায় এই কাজকে প্রোগ্রামিং অ্যানালিসিস (Programming Analysis) বলে।

উদাহরণ : আমাদের পূর্ব আলোচিত ক্যাশমেমো তৈরীর সফট্ওয়্যারের কথাই ধরুন না। এই সফট্ওয়্যারটি যদি আপনি কোনো প্রোগ্রামারকে লিখতে দেন, তাহলে তাঁকে কেবলমাত্র "ক্যাশ-মেমো তৈরীর সফট্ওয়্যার লিখে দিন" এই কথাটুকু বললেই কাজ হবে না। তিনি আপনার কাছে জানতে চাইবেন সফট্ওয়্যারটি সঠিক কি কি কাজ করবে। অর্থাৎ আপনাকে তখন বলে দিতে হবে "সফট্ওয়্যারটি এমনভাবে লিখবেন, যাতে ক্যাশমেমোর ওপরে এবং নিচে বর্ডার থাকে। ওপরের বর্ডারের নিচে লেখা থাকবে Cash Memo এবং তার নিচেও একটি ছোট বর্ডার থাকবে। ছোট বর্ডারের নিচে লেখা থাকবে ক্রেতার নাম, ঠিকানা, পণ্যের সংখ্যা, নাম, দাম। এরপর থাকবে মোট মূল্যের ওপর ৪% হারে নির্ধারিত সেলস্ ট্যাক্স। মোট মূল্য ও সেলস ট্যাক্স যোগ করে ক্রেতার দেয় মূল্যটি লেখা থাকবে সব শেষে। তাছাড়া সফট্ওয়্যারটি RUN করার পর কম্পিউটার প্রতিটি ভেরিয়েবলের মান জানতে চাওয়ার আগে ব্যবহারকারীকে বলে দেবে কোন ভেরিয়েবলের মান সে জানতে চাইছে।"

এই হলো মোটামুটি ক্যাশমেমো প্রোগ্রামের কাজকর্ম সম্বন্ধে প্রাথমিক তথ্য। কম্পিউটার পরিভাষায় একে প্রবলেম ডেফিনেশন (Problem Definition) বলে।

সফট্ওয়্যারের উদ্দেশ্য অর্থাৎ (Problem definition) বুঝে ফেলার পর প্রোগ্রামিং অ্যানালিসিসের অন্তর্গত দ্বিতীয় কাজ হবে, প্রবলেম ডেফিনেশন অনুযায়ী সফট্ওয়্যারটি তৈরী করতে যাবতীয় প্রয়োজনীয় তথ্য ও ফর্মুলা সংগ্রহ করা এবং সেই তথ্য ও ফর্মুলা

পরপর ব্যবহার করে কিভাবে সফট্ওয়্যারটি লেখা হবে তার একটি কাঠামো খাড়া করা। কম্পিউটার পরিভাষায় একে বলা হয় অ্যালগরিদম (Algorithem)



অ্যালরিদম

ক্যাশমেমো প্রোগ্রামের কথাই যদি ধরা যায় তবে তার অ্যালগরিদমটি হবে এইরকম—

(1) ক্রেতার নাম ঠিকানা জিজ্ঞাসা কর

(2) পণ্যের পরিমাণ, নাম এবং দাম জিজ্ঞাসা কর

(3) পণ্যের পরিমাণের সঙ্গে দাম গুণ কর

(4) সেই গুণফলের ওপর সেলস্ ট্যাক্স নির্ণয় কর

(5) সেলস্ ট্যাক্স এবং গুণফল যোগ করে সেটি মোট মূল্য বলে গণ্য কর

(6) ক্যাশমেমোর বর্ডার ও হেডিং ছাপো

(7) ক্রেতার নাম, ঠিকানা, পণ্যের নাম, দাম, পরিমাণ ইত্যাদি ছাপো

(8) সেলস্ ট্যাক্সের পরিমাণ ছাপো

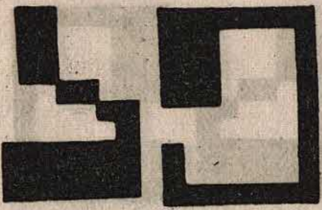
(9) মোট মূল্য ছাপো

(10) শেষ করো

অ্যালগরিদম সম্পূর্ণ হলে প্রেগ্রামিং অ্যানালিসিসের অন্তর্গত শেষ কাজ হলো ফ্লোচার্ট তৈরী করা।

ফ্লোচার্ট

ফ্লোচার্ট (Flowchart) বলতে বোঝায় সফট্ওয়্যার লেখার সময় প্রোগ্রামার যে যুক্তিগ্রাহ্য পথ (logical path) ধরে এগোবেন, তারই সরলীকৃত নক্সা (graphical representation)। অর্থাৎ প্রোগ্রামটি ঠিক কিভাবে কাজ করবে সেটি ফ্লোচার্টের মাধ্যমে ছবি এঁকে দেখানো হয়। ফ্লোচার্ট থেকে প্রোগ্রামটির কার্যপ্রণালী এতো সহজভাবে বোঝা যায় যে, যেকোনো কম্পিউটার প্রোগ্রামিং সম্বন্ধে অনভিজ্ঞ ব্যক্তিও ফ্লোচার্ট দেখে তার মানে বুঝতে পারেন। উদাহরণ



ফ্লোচার্ট

চিত্র ৬৫

স্বরূপ ক্যাশমেমো প্রোগ্রামের ফ্লোচার্টটাই দেখুন, দেখবেন আপনার যদি আগে থেকে ফ্লোচার্ট পড়ার নিয়ম কানুন জানা নাও থাকে তাহলেও আপনি এই প্রোগ্রামটির কার্যপ্রণালী স্পষ্ট বুঝতে পারবেন।

ফ্লোচার্টটিতে দেখা যাচ্ছে বিভিন্ন নির্দেশ ও তথ্যগুলি বিভিন্ন আকৃতির খোপের মধ্যে বসানো আছে। এই খোপগুলির কিন্তু নির্দিষ্ট অর্থ আছে। বস্তুত এরাই হলো ফ্লোচার্টে ব্যবহৃত বিভিন্ন সাংকেতিক চিহ্ন। তাছাড়া কোন খোপের পর কোন খোপে যেতে হবে সেটিও তীর চিহ্নের মাধ্যমে দেখানো আছে। এর নাম হলো ফ্লো লাইন (Flow লাইন)। ফ্লো লাইনও ফ্লোচার্টের একটি সাঙ্কেতিক চিহ্ন।

ক্যাশমেমো ছাপানোর ফ্লোচার্ট

A = প্রোগ্রামের শুরু

B = প্রোগ্রামের অন্তর্গত ভেরিয়েবল গুলির মান জানতে চাওয়া অর্থাৎ ক্রেতার নাম, ঠিকানা, পণ্যের নাম, পরিমাণ ও দাম জিজ্ঞাসা করা

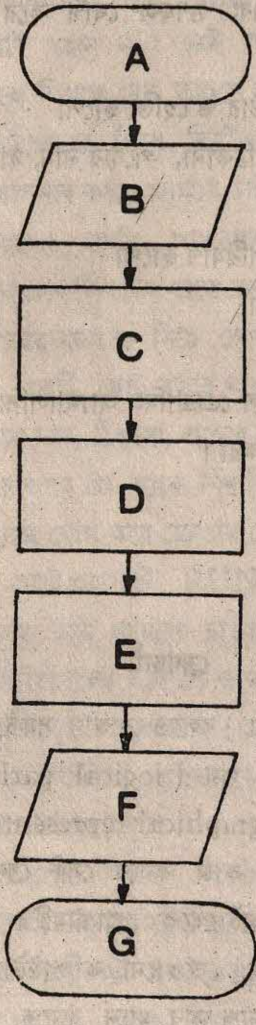
C = পণ্যের পরিমাণ ও একক পরিমাণের দাম গুণ করে গুণফলকে আর একটি নতুন ভেরিয়েবলের মান হিসেবে নির্দিষ্ট করা

D = এই নতুন ভেরিয়েবলের মানের ওপর ৪% হারে সেলস্ ট্যাক্স নির্ধারণ করা

E = সেলস্ ট্যাক্স এবং নতুন ভেরিয়েবলের মান একত্রে যোগ করে মোট মূল্য নির্ণয় করা

F = ক্যাশমেমোতে ক্রেতার নাম, ঠিকানা, পণ্যের নাম, পরিমাণ, দাম, সেলস্ ট্যাক্সের হার এবং মোট মূল ছাপিয়ে দেওয়া

G = প্রোগ্রামটি শেষ করা



১৫
ফ্লোচার্ট

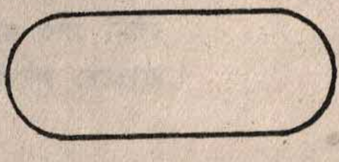
এটি প্রোগ্রামের যুক্তিগ্রাহ্য পথ নির্দেশ করে (যে পথ ধরে C.P.U ধাপে ধাপে সমস্যা সমাধান করতে অগ্রসর হয়)। ফ্লোচার্ট লিখতে শেখার আগে আমাদের তাই ফ্লোচার্টের সঙ্কেতগুলির সঙ্গে পরিচিত হতে হবে।

ফ্লোচার্টের সাঙ্কেতিক চিহ্ন

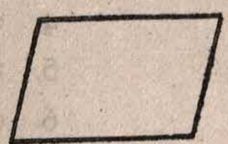
ফ্লোচার্ট আঁকতে খুব বেশী সাঙ্কেতিক চিহ্ন প্রয়োজন হয় না। সর্বসাকুল্যে মাত্র ছটি সঙ্কেতই যথেষ্ট। এগুলি ছাড়াও আরো অনেকগুলি সাঙ্কেতিক চিহ্ন ফ্লোচার্টের জন্য নির্দিষ্ট আছে, তবে সেগুলি বেশীর ভাগ ক্ষেত্রে বিশেষ প্রয়োজন হয় না।

এই সঙ্কেতগুলি সৃষ্টি করেছেন অ্যামেরিকান গ্যাশনাল স্টাণ্ডার্ডস্ ইন্সটিটিউট (ANSI) এবং বর্তমানে ANSI নির্দিষ্ট সঙ্কেতগুলিই আন্তর্জাতিকভাবে স্বীকৃত এবং ব্যবহৃত ফ্লোচার্ট সঙ্কেত।

Terminal Symbol—টার্মিনাল সিম্বলের বাংলা করলে দাঁড়ায় প্রান্তিক সঙ্কেত। অনেকটা ক্যাপস্যুল আকৃতির খোপটি ফ্লোচার্টের শুরু এবং শেষ চিহ্নিত করে। (যেমন BASIC-এর NEW এবং END)।



Imput/Output Symbol—ভেরিয়েবলের মান জানতে চাওয়া কিম্বা আউটপুট ডিভাইসের মাধ্যমে ভেরিয়েবলের মান জানিয়ে দেওয়ার নির্দেশ বোঝাতে ইনপুট/আউটপুট সিম্বল ব্যবহার করা হয়। (যেমন INPUT, READ/DATA এবং WRITE স্টেটমেন্ট)

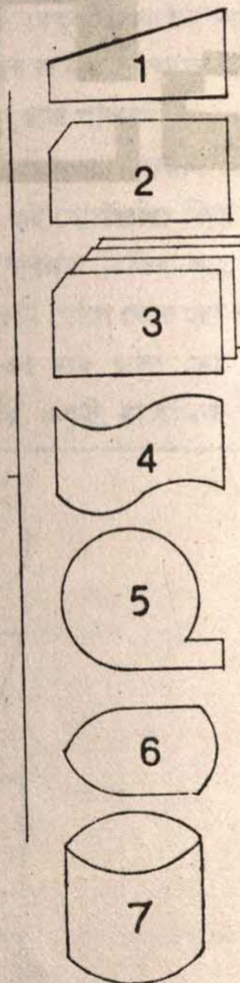


১৩

ফ্লোচার্ট

ANSI প্রবর্তিত ফ্লোচার্ট সংকেত গুলির মধ্যে ইনপুট / আউটপুট সংকেতটির আবার অনেকগুলি বিভাগ আছে। এখানে 7টি বহুল প্রচলিত ইনপুট / আউটপুট সংকেত দেখানো হলো।

যখন কোনো প্রোগ্রামে তথ্য পড়া বা লেখার জন্য একাধিক ভিন্ন ভিন্ন ইনপুট বা আউটপুট ডিভাইস ব্যবহার করা হয়, তখনই এই সংকেত গুলি ব্যবহার করা হয়। নইলে সাধারণ সমস্ত ক্ষেত্রে আগের পাতায় দেওয়া সংকেতটিই ব্যবহার হয়ে থাকে।



চিত্র ৬৬

বিভিন্ন I/0 ডিভাইসের ফ্লোচার্ট সংকেত

1. ম্যানুয়াল (Manual) ইনপুট
2. পাঞ্চড্ (Punched Card) কার্ড
3. নির্দিষ্ট ক্রমে সাজানো একাধিক কার্ড বা কার্ড ডেক্ (Deck of Cards)
4. পাঞ্চড্ টেপ (Punched tape)
5. ম্যাগনেটিক টেপ (Magnetic tape)
6. যে কোনো ধরনের ডিসপ্লে (Display)
7. ফ্লপি কিম্বা উইনচেস্টার ডিস্ক (Magnetic disk)

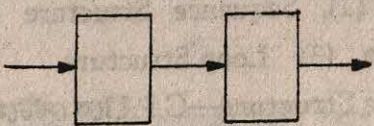
ফ্লোচার্ট

Processing Symbol—আয়তক্ষেত্রাকার খোপটির নাম প্রসেসিং সিম্বল। সফট্ওয়্যারের অন্তর্গত সমস্ত গাণিতিক হিসেব নিকেশ এবং ভেরিয়েবলের মান নির্দিষ্টকরণের নির্দেশ এই প্রসেসিং সিম্বলের ভিতরে লেখা হয়।

(যেমন BASIC ভাষার লাইব্রেরী ফংশন ও LET স্টেটমেণ্ট)

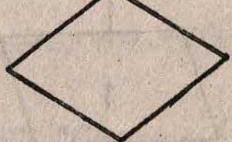


Flow line—ফ্লোচার্টে কোন্ সঙ্কেতের পর কোন্ সঙ্কেত পড়তে হবে সেটি দেখিয়ে দেয় তীর চিহ্ন আকৃতির ফ্লো লাইন। প্রোগ্রাম লেখা হবার পর C.P.U ফ্লো লাইন অনুসরণ করে সমস্যাটি ধাপে ধাপে সমাধান করে।



Decision Symbol—কোনো বিশেষ অবস্থার ওপর নির্ভর করে সফট্ওয়্যারকে দুটি প্রশাখায় বিভক্ত করার নির্দেশ বোঝাতে ডিশিশন সিম্বল ব্যবহার করা হয়।

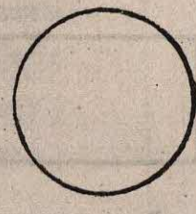
(যেমন IF/THEN স্টেটমেণ্ট)





ফ্লোচার্ট

Connecter Symbol—ফ্লোচার্ট আঁকার সময় প্রায়ই জটিলতা এড়ানোর জন্য সমস্ত সঙ্কেত পরস্পরের সঙ্গে সরাসরি ফ্লোলাইন দ্বারা যুক্ত করা হয় না। এই সব পরস্পর বিচ্ছিন্ন অংশগুলির মধ্যে সংযোগ বোঝাতে ব্যবহার করা গোলাকৃতি সঙ্কেত, যার নাম কানেক্টর সিম্বল।



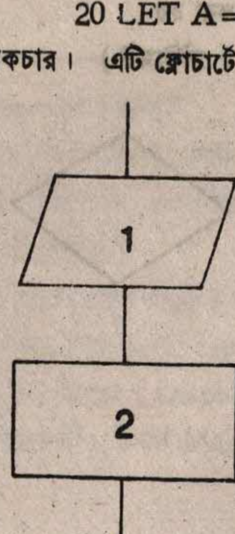
ফ্লোচার্ট সঙ্কেতের ব্যবহার

সফট্‌ওয়্যার লিখতে গিয়ে দেখা গেছে, যে কোনো সমস্যাকেই তিনটি মূল প্রোগ্রামিং পদ্ধতিতে ভেঙে নিয়ে সমাধান করা যায়। এই তিনটি পদ্ধতি হলো. (1). Sequence Structure (2) Selection Structure এবং (3) Loop Structure.

Sequence Structure—C.P.U-কে স্টেটমেন্ট নাম্বার অনুসারে পরপর নির্দেশ পালন করতে বলাকেই সিকোয়েন্স স্ট্রাকচার বলা হয়। ফ্লোচার্টে যে কোনো পরপর ফ্লোলাইন দ্বারা সংযুক্ত সঙ্কেতই সিকোয়েন্স স্ট্রাকচার বোঝায়। যেমন

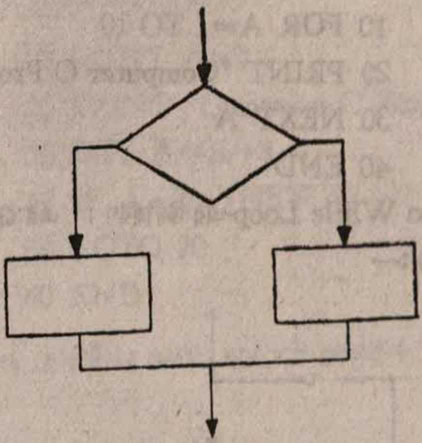
```
10 READ A
20 LET A=A+1
```

একটি সিকোয়েন্স স্ট্রাকচার। এটি ফ্লোচার্টে আঁকলে দাঁড়ায় এইরকম



Selection Structure—ভেরিয়েবলের মানের ওপর নির্ভর করে প্রোগ্রামকে দুটি প্রশাখায় (Branch) বিভক্ত করাকে বলা হয় সিলেক্‌শান স্ট্রাকচার।

যেমন 10 IF A=0 THEN 350
20 LET A=50
350 LET A=60 ফ্লোচার্টে সিলেক্‌শান স্ট্রাকচার আঁকা হয় এই ভাবে।



Loop Structure—প্রোগ্রামের কোন অংশ থেকে C.P.U. নিয়ন্ত্রণকে পুনরায় তার পূর্বের অংশে ফিরিয়ে নিয়ে যাওয়াকে বলা হয় লুপ স্ট্রাকচার। লুপ স্ট্রাকচারকে আবার তিনভাগে ভাগ করা যেতে পারে :—

(a) এণ্ডলেস লুপ (Endless Loop) (b) ডু হোয়াইল লুপ (Do While Loop) (c) ডু আনটিল লুপ (Do Untill Loop).

Endless Loop—এই রকম লুপের সঙ্গে আমরা ইতিপূর্বেই পরিচিত হয়েছি।

যেমন—10 PRINT "Computer O Programming"
20 GOTO 10

এই ছোট্ট প্রোগ্রামটি একটি এণ্ডলেস লুপ স্ট্রাকচার। ফ্লোচার্ট আঁকা হলে দাঁড়াবে এই রকম :—

ফ্লোচার্ট

Do While Loop—ডু হোয়াইল লুপও আমাদের পরিচিত। এই লুপ থেকে C.P.U নিয়ন্ত্রণকে বার করে আনতে লুপের শুরুতেই একটি সিলেক্‌শন স্ট্রাকচার থাকে। ডু হোয়াইল লুপের বৈশিষ্ট্য হল লুপ কাউন্টার ভেরিয়েবলের মান যতক্ষণ না একটি পূর্ব নির্দিষ্ট মান অতিক্রম করে, C.P.U নিয়ন্ত্রণটি ততক্ষণ ওই লুপের মধ্যে ঘুরতে থাকে।

```
10 FOR A=1 TO 10
20 PRINT "Computer O Programming."
30 NEXT A
40 END
```

হল একটি Do While Loop-এর উদাহরণ। এর ফ্লোচার্টের চেহারা হবে এই রকম :—



এই একই Do While Loop-কে আমরা IF, THEN-এর সাহায্যেও লিখতে পারি—

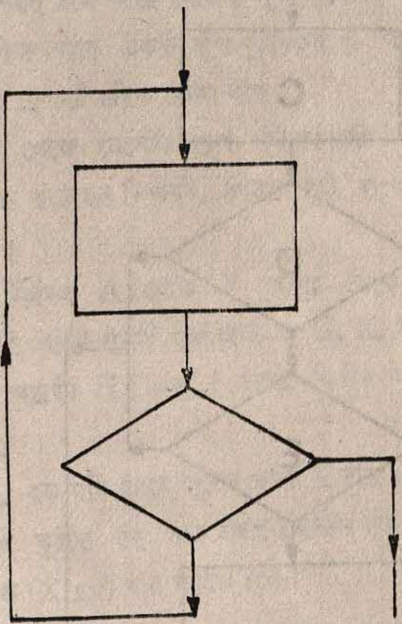
```
10 LET A=1
20 IF A=10 THEN 60
30 PRINT "Computer 0 programming"
40 LETA=A+1
```

```
50 GOTO 20
60 END
```

Do Untill Loop—ডু আন্টিল লুপের চরিত্র কিছুটা ডু হোয়াইল লুপেরই মতো। তবে এই লুপে সিলেকশন স্ট্রাকচারটি থাকে একেবারে শেষে। আগের প্রোগ্রামটিই যদি Do Untill Loop স্ট্রাকচারে লেখা হয় তাহলে সেটি হবে—

```
10 PRINT A=1
20 PRINT "Computer O Programming."
30 LET A=A+1
40 IF A=10 THEN 60
50 GOTO 20
60 END
```

ডু আন্টিলের ফ্লোচার্টটির চেহারা হবে এই রকম :—

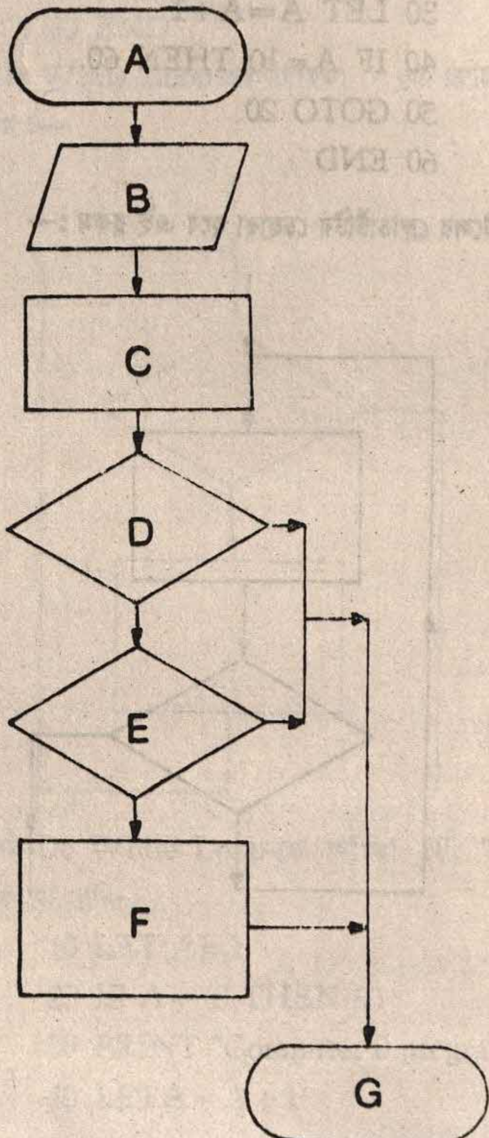


ফ্লোচার্ট

এবার দু-একটি সহজ প্রোগ্রামের ফ্লোচার্ট এঁকে ব্যাপারটি আরো পরিষ্কারভাবে বুঝে নেওয়া যাক।

অ্যাপ্লিকেশন ফর্ম বাছাইয়ের ফ্লোচার্ট

এই প্রোগ্রামটির সফট্ওয়্যার আমরা গত পরিচ্ছেদেই লিখে ফেলেছি। এবার ওই প্রোগ্রামটির ফ্লোচার্টটি কেমন হবে সেটি দেখা যাক :—



ফ্লোচার্টটির সঙ্গে আমাদের একটু আগে শেখা নিয়মগুলি মিলিয়ে দেখুন, দেখবেন এই ফ্লোচার্টটি আঁকতে কেবলমাত্র সিকোয়েন্স স্ট্রাকচার ও সিলেক্শন স্ট্রাকচার ব্যবহার করতে হয়েছে। অর্থাৎ এর থেকে প্রমাণ হচ্ছে অন্তত এই প্রোগ্রামটিকে সমাধান করতে কেবল ওই দুটি স্ট্রাকচারই যথেষ্ট। বস্তুতপক্ষে সিকোয়েন্স, সিলেক্শন এবং লুপ স্ট্রাকচারের সাহায্যেও যে কোনো রকম প্রোগ্রামই সমাধান করা যায়। আসুন এবার আর একটি প্রোগ্রামের ফ্লোচর্টে আঁকা যাক যাতে সব রকম স্ট্রাকচারই ব্যবহার করা হয়েছে।

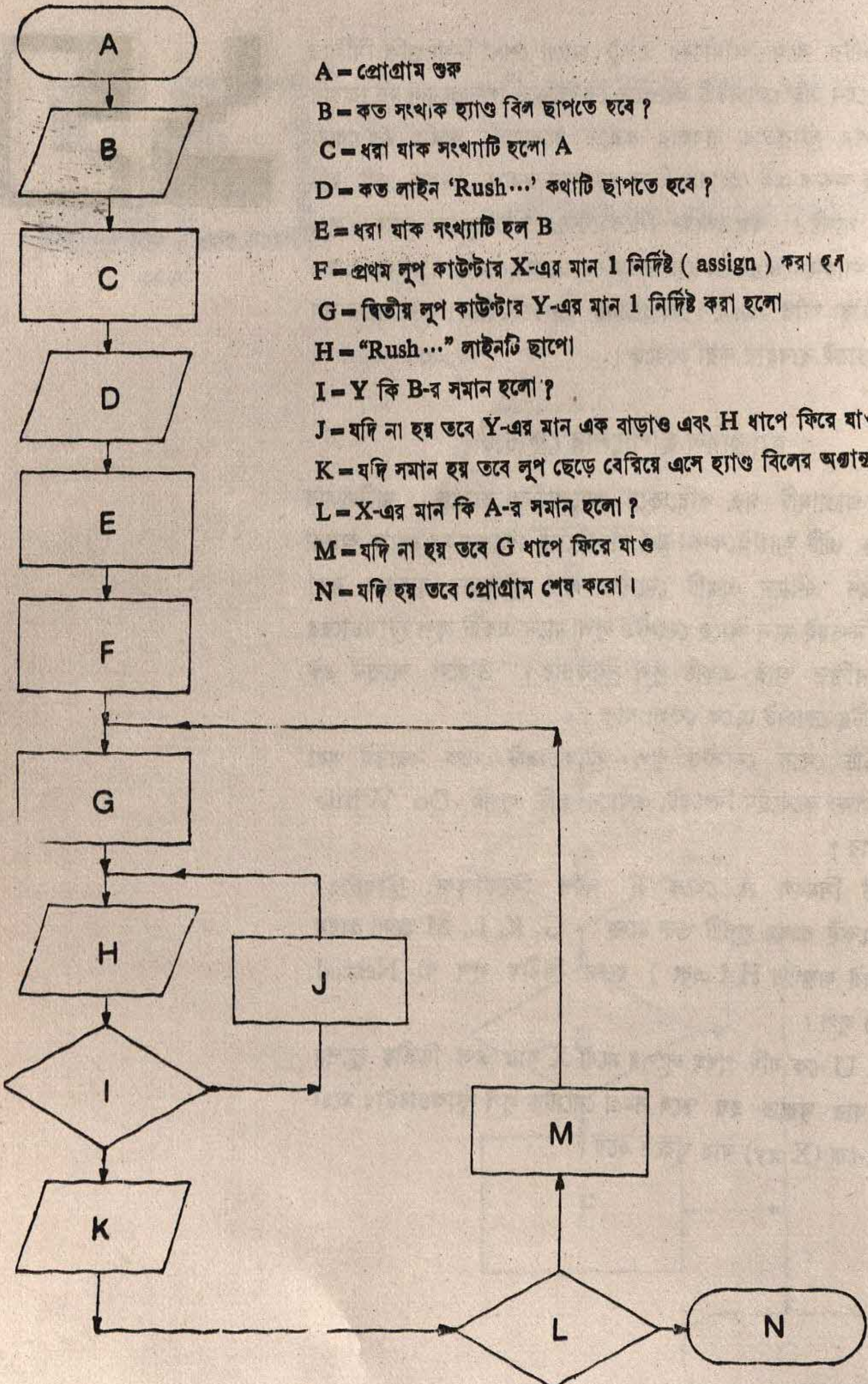
হ্যাণ্ডবিল ছাপানোর ফ্লোচার্ট

এই প্রোগ্রামটি গত পরিচ্ছেদে আলোচিত হয়েছে। জটিলতার দিক থেকে এটি অ্যাপ্লিকেশন ফর্ম বাছাইয়ের প্রোগ্রামের থেকে আরো জটিল কারণ এখানে একটি নেস্টেড লুপ স্ট্রাকচার ব্যবহার করা হয়েছে। নিশ্চয়ই মনে আছে নেস্টেড লুপ মানে একটি লুপ স্ট্রাকচারের ভিতরে অবস্থিত আর একটি লুপ স্ট্রাকচার? তাহলে আসুন এই সফ্‌টওয়্যারটির ফ্লোচার্ট এঁকে ফেলা যাক।

ফ্লোচার্টটি থেকে নেস্টেড লুপ স্ট্রাকচারটি এক নজরেই ধরা পড়ছে। লক্ষ্য করেছেন নিশ্চয়ই, এখানে দুটি লুপই Do While লুপ স্ট্রাকচার?

ফ্লোচার্ট সিমবল A থেকে F পর্যন্ত সিকোয়েন্স স্ট্রাকচার। এরপর থেকেই প্রথম লুপটি শুরু হচ্ছে। G, K, L, M হলো প্রথম লুপ। এরই অন্তর্গত H I এবং J হলো দ্বিতীয় লুপ বা Nested (নেস্টেড) লুপ।

C. P. U-কে যদি প্রথম লুপের মধ্যে X বার এবং দ্বিতীয় লুপের মধ্যে Y বার ঘুরতে হয় তবে সমগ্র নেস্টেড লুপ স্ট্রাকচারটার মধ্যে C. P. U.-কে (X xy) বার ঘুরতে হবে।



A = প্রোগ্রাম শুরু

B = কত সংখ্যক হ্যাণ্ড বিল ছাপতে হবে ?

C = ধরা যাক সংখ্যাটি হলো A

D = কত লাইন 'Rush…' কথাটি ছাপতে হবে ?

E = ধরা যাক সংখ্যাটি হল B

F = প্রথম লুপ কাউন্টার X-এর মান 1 নির্দিষ্ট (assign) করা হল

G = দ্বিতীয় লুপ কাউন্টার Y-এর মান 1 নির্দিষ্ট করা হলো

H = "Rush…" লাইনটি ছাপো

I = Y কি B-র সমান হলো ?

J = যদি না হয় তবে Y-এর মান এক বাড়াও এবং H ধাপে ফিরে যাও

K = যদি সমান হয় তবে লুপ ছেড়ে বেরিয়ে এসে হ্যাণ্ড বিলের অন্যান্য অংশ ছাপো ।

L = X-এর মান কি A-র সমান হলো ?

M = যদি না হয় তবে G ধাপে ফিরে যাও

N = যদি হয় তবে প্রোগ্রাম শেষ করো ।

ফ্লোচার্ট আঁকার নিয়ম

আমরা কিন্তু ফ্লোচার্ট আঁকার নিয়ম না শিখেই দুদুটো প্রোগ্রামের ফ্লোচার্ট এঁকে ফেললাম। আসল কথা কি জানেন? ফ্লোচার্ট আঁকার কোনো বাঁধা ধরা নিয়মই নেই! কোনো একটি প্রোগ্রামের ফ্লোচার্ট আঁকার সময় সেটি কত বিশদভাবে আঁকতে হবে, সে বিষয়ে আন্তর্জাতিকভাবে কোনোই নির্দিষ্ট মান বেঁধে দেওয়া নেই। ফ্লোচার্টে প্রোগ্রামের কত খুঁটিনাটি ধাপ দেখানো হবে সেটি একান্তভাবে প্রোগ্রামারের ইচ্ছার ওপর নির্ভর করে। তাছাড়া বিভিন্ন স্ট্রাকচার-গুলি ফ্লোলাইন দিয়েজোড়ার সময় সেগুলি আনুভূমিক (Horizontal) ভাবে নাকি উল্লম্ব (Vertical) ভাবে সাজানো হবে, সেটাও নির্ভর করে প্রোগ্রামারের মর্জির ওপর।

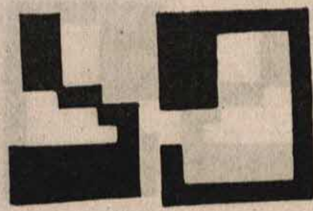
এক কথায় ফ্লোচার্টটি সফট্ওয়্যার রচনার কাঠামোর রচনার কাজ করে। মৃৎশিল্পীরা যেমন খড়ের কাঠামোর ওপর মাটি চড়িয়ে মূর্তির পূর্ণাঙ্গ রূপ দেন, ঠিক তেমনি কম্পিউটার প্রোগ্রামারও ফ্লোচার্ট কাঠামোর ওপর হাই লেভেল ল্যাংগোয়েজের প্রলেপ চড়িয়ে সফট্ওয়্যার রচনা সম্পূর্ণ করেন।

ফ্লোচার্ট ব্যবহার করে সফট্ওয়্যার ডিজাইন করা

এবার দেখা যাক বড় ও জটিল প্রোগ্রাম কি ভাবে ফ্লোচার্টের সাহায্যে সহজে ডিজাইন করা যায়। যে প্রোগ্রামটি আমরা ডিজাইন করতে যাচ্ছি, তার নাম মিনি ডেটা বেস (Data Base) প্রোগ্রাম অথবা কম্পিউটারাইজড্ নোট বুক প্রোগ্রাম। এই প্রোগ্রামটিকে আমরা পেশাদার প্রোগ্রামারের কায়দায় প্রথমে প্রবলেম ডিফিনেশন করে, অ্যালগরিদ্‌ম্ লিখে, ফ্লোচার্ট এঁকে তারপর BASIC ভাষায় লিখবো।

কম্পিউটারাইজড্ নোট বুক সফট্ওয়্যার ডিজাইনিং

প্রয়োজনীয় তথ্য লিখে রাখতে আমরা নোটবুক কিম্বা ডায়রী ব্যবহার করি। অথচ এ ধরনের কাছে কম্পিউটার আরো অনেক বেশী দক্ষ। কম্পিউটারের বিশাল সেকেণ্ডারী মেমরি অর্থাৎ বাড়তী স্মৃতি ভাণ্ডারে ইচ্ছা করলে লাখ লাখ তথ্য লিখে রাখা যায়। এভাবে তথ্য সংরক্ষণ করতে সাধারণ ডায়রীর থেকে জায়গা বহু বহু গুণ কম লাগে আর তথ্যের গোপনীয়তাও বহুগুণ বেড়ে যায়। তাছাড়া ওই সব



ডেটা বেস সফট্ওয়্যার

অফুরন্ত স্মৃতিভাণ্ডার এবং অকল্পনীয় দ্রুততার সঙ্গে তথ্য খুঁজে আনার ক্ষমতার জন্য ডেটা বেস হিসেবে কম্পিউটারের গুরুত্ব অপরিসিম। বিদেশে অনেক ডেটা বেস কোম্পানী আছেন যাঁরা হাজার হাজার বিভিন্ন বিষয়ের তথ্য তাঁদের কম্পিউটার ডেটা ব্যাঙ্কে জমিয়ে রাখেন। ওই সব কোম্পানীর গ্রাহকেরা মডেমের সাহায্যে কোম্পানীর কম্পিউটার ডেটা ব্যাঙ্ক থেকে যখন খুশি সেই তথ্য পড়ে নিতে পারে। এক কথায় 24 ঘণ্টা ধরে পৃথিবীর প্রায় যাবতীয় বিষয়ের সর্বশেষ খবরা খবর পাওয়ার ব্যবস্থা হল এই ডেটা বেস। বর্তমানে এরই একটা বিশেষ রূপ ভারতেও Teletext নামে চালু হয়েছে।



ডেটা বেস প্রবলেম ডেফিনেশন

স্মৃতিতে রক্ষিত তথ্য পড়ার দরকার হলে কম্পিউটার অত্যন্ত কম সময়ের মধ্যে স্মৃতি ভাণ্ডারে রক্ষিত তথ্য খুঁজে বার করে আনতে পারে, যা ডায়রী বা নোটবুকের মধ্যে থেকে কখনোই অত দ্রুত খুঁজে বার করা সম্ভব নয়। এই ধরনের কাজেই বর্তমানে বড় বড় প্রতিষ্ঠানের কম্পিউটারগুলি সব থেকে বেশী ব্যবহার করা হয়। কম্পিউটার পরিভাষায় এই কাজের প্রোগ্রামকে বলা হয়। ডেটা বেস প্রোগ্রাম। আসল ডেটা বেস প্রোগ্রাম লিখতে Enhanced BASIC-এর প্রয়োজন হয়, কিন্তু আমরা যেহেতু Enleanced BASIC শিখছি না, তাই মূল BASIC-এর সাহায্যেই একটি ছোট ডেটা বেস প্রোগ্রাম আমরা তৈরী করবো। এই প্রোগ্রামটির প্রবলেম ডেফিনেশন নিচে দেওয়া হলো—

প্রবলেম ডেফিনেশন

(1) সফট্‌ওয়্যারটি সম্পূর্ণ মূল BASIC কী ওয়ার্ড ব্যবহার করে লিখতে হবে।

(2) বিশেষ সংকেত বাক্যের (Pass Word) সাহায্যে কম্পিউটারে সঞ্চিত তথ্যের গোপনীয়তার ব্যবস্থা করতে হবে।

(3) সঞ্চিত তথ্য কেউ চুরি করতে চাইলে তাকে বোকা বানিয়ে বিফল করতে হবে।

(4) স্মৃতি ভাণ্ডারে A থেকে Z, এই 26টি ইংরিজি অক্ষরের জন্য 26টি আলাদা আলাদা ডেটা ব্যাঙ্ক (Data Bank) তৈরী করতে হবে।

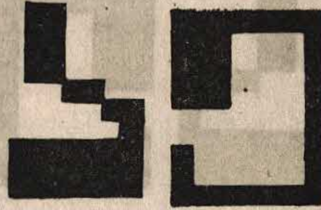
(5) ব্যবহারকারী কোন্ অক্ষরের জন্য নির্দিষ্ট ডেটা ব্যাঙ্কের তথ্য পড়তে চাইছে তা জানতে চাইতে হবে।

(6) তথ্য যোগান দেবার পর প্রশ্ন করে জেনে নিতে হবে ব্যবহারকারী আর অন্য কোনো ডেটা ব্যাঙ্কের তথ্য পড়তে চায় কিনা।

(7) যদি উত্তর হ্যাঁ হয় তবে পুনরায় 5 নম্বর ধাপ থেকে শুরু করতে হবে।

(8) যদি উত্তর না, হয় তবে ধন্যবাদ জানিয়ে বিদায় নিতে হবে।

প্রবলেম ডেফিনেশন জেনে নেওয়ার পর প্রোগ্রামটির অ্যালগরিদ্‌মটি লিখলে দাঁড়াবে এই রকম :—



ডেটা বেস অ্যালগরিদম

অ্যালগরিদম্

(1) সঙ্কেত বাক্য (Pass Word) জিজ্ঞাসা কর।

(2) সঙ্কেত বাক্য ভুল হলে বোকা বানানোর ব্রাঞ্চ স্ট্রাকচার 4 নম্বরে যাও।

(3) সঙ্কেত বাক্য সঠিক হলে তথ্য জানানোর ব্রাঞ্চ স্ট্রাকচার 6 নম্বরে যাও।

(4) ডেটা ব্যাঙ্কের অদ্যাক্ষর জানতে চাও।

(5) যে কোনো ইনপুট দেওয়া হলে পুনরায় 4 নম্বরে ফিরে যাও।

(6) ডেটা ব্যাঙ্কের অদ্যাক্ষর জানতে চাও।

(7) সেই অদ্যাক্ষরের ডেটা ব্যাঙ্কের তথ্য মনিটর স্ক্রিনে ফুটিয়ে তোল।

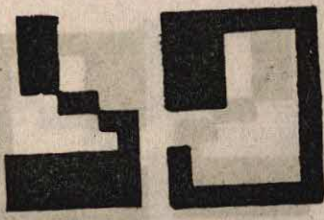
(8) প্রোগ্রামারকে প্রশ্ন করো আরো তথ্য প্রয়োজন কিনা।

(9) যদি উত্তর হ্যাঁ হয় তবে 6 নম্বরে ফিরে যাও।

(10) যদি উত্তর না হয় তবে ধন্যবাদ জানিয়ে শেষ করো।

কম্পিউটারাইজড্ নোট বুকের ফ্লোচার্ট

অ্যালগরিদম থেকে ফ্লোচার্টটা এঁকে ফেলতে হবে। এর জন্য প্রথমে বারবার অ্যালগরিদমটা পড়ে পুরো ব্যাপারটি সম্পূর্ণ পরিষ্কার ভাবে বুঝে নিতে হবে। তারপর অ্যালগরিদমে ঠিক যে ধাপের পর যে ধাপ লেখা আছে, সেই ধারাবাহিকতা বজায় রেখে উপযুক্ত ফ্লোচার্ট সিম্বল এবং ফ্লোলাইনের সাহায্য নিয়ে ফ্লোচার্ট আঁকতে হবে। প্রথমবার ফ্লোচার্টে প্রোগ্রামের সমস্ত খুঁটিনাটি বিষয় না দেখালেও চলবে। একে বলা হবে Macro Flow chart (ম্যাক্রো ফ্লোচার্ট)। ম্যাক্রো ফ্লোচার্ট আঁকা হয়ে গেলে তারপর তাতে ক্রমশ প্রোগ্রামের খুঁটিনাটি বিষয়গুলি জুড়তে হবে। এই ভাবে তৈরী হবে Micro flow chart (মাইক্রো ফ্লোচার্ট)। মাইক্রো চার্ট থেকে সরাসরি প্রোগ্রাম লিখে ফেলা যাবে। ২৫১ পাতায় যে ফ্লোচার্টটি দেওয়া হলো সেটি ম্যাক্রো এবং মাইক্রো ফ্লোচার্টের মাঝামাঝি ফ্লোচার্ট বলা যেতে পারে। বোঝবার সুবিধার জন্য প্রোগ্রামের সমস্ত খুঁটিনাটি ধাপগুলি দেখানো হয়নি।



ডেটা বেস ফ্লোচার্ট

কম্পিউটারাইজড নোট বুক প্রোগ্রামের ফ্লোচার্টে ব্যবহৃত বিভিন্ন সিমবলের অর্থ বিশ্লেষণ

A = প্রোগ্রামের শুরু

B = সংকেত বাক্য ?

C = সংকেত বাক্য সঠিক ?

D = যদি সঠিক না হয় তবে প্রোগ্রামারকে বোকা বানানোর জন্য ডেটা ব্যাঙ্কের নাম জিজ্ঞাসা কর

E = সংকেত বাক্য পড় এবং আবার Dতে ফিরে যাও

F = সংকেত বাক্য সঠিক হলে ডেটা ব্যাঙ্কের নাম জিজ্ঞাসা কর

G = নামটি কি 'A' ?

H = যদি 'A' হয় তবে 'A' ব্যাঙ্কের তথ্য ছাপো

I = যদি 'A' না হয় তবে নামটি কি 'B' ?

J = যদি B হয় তবে 'B' ব্যাঙ্কের তথ্য ছাপো

··· = যদি B না হয় তবে একই ভাবে পরবর্তি ধাপে চলে যাও।

[ফ্লোচার্টের জটিলতা এবং আকার কমানোর জন্য 26টি ইংরিজি হরফের প্রত্যেকটির ক্ষেত্রে 'নামটি কি ' ' হরফ' এবং 'তথ্য ছাপো' ব্যাপারটি দেখানো হয়নি। তবে প্রতিটি ক্ষেত্রে ফ্লোচার্টটি হুবহু একই রকম হবে। তাই এই অংশগুলি ডট্‌ডট্ চিহ্ন দিয়ে দেখানো হয়েছে]

H, J···L, N ইত্যাদি অর্থাৎ ডেটা ব্যাঙ্কের তথ্য ছাপার পর ফ্লোলাইন যাচ্ছে O-তে।

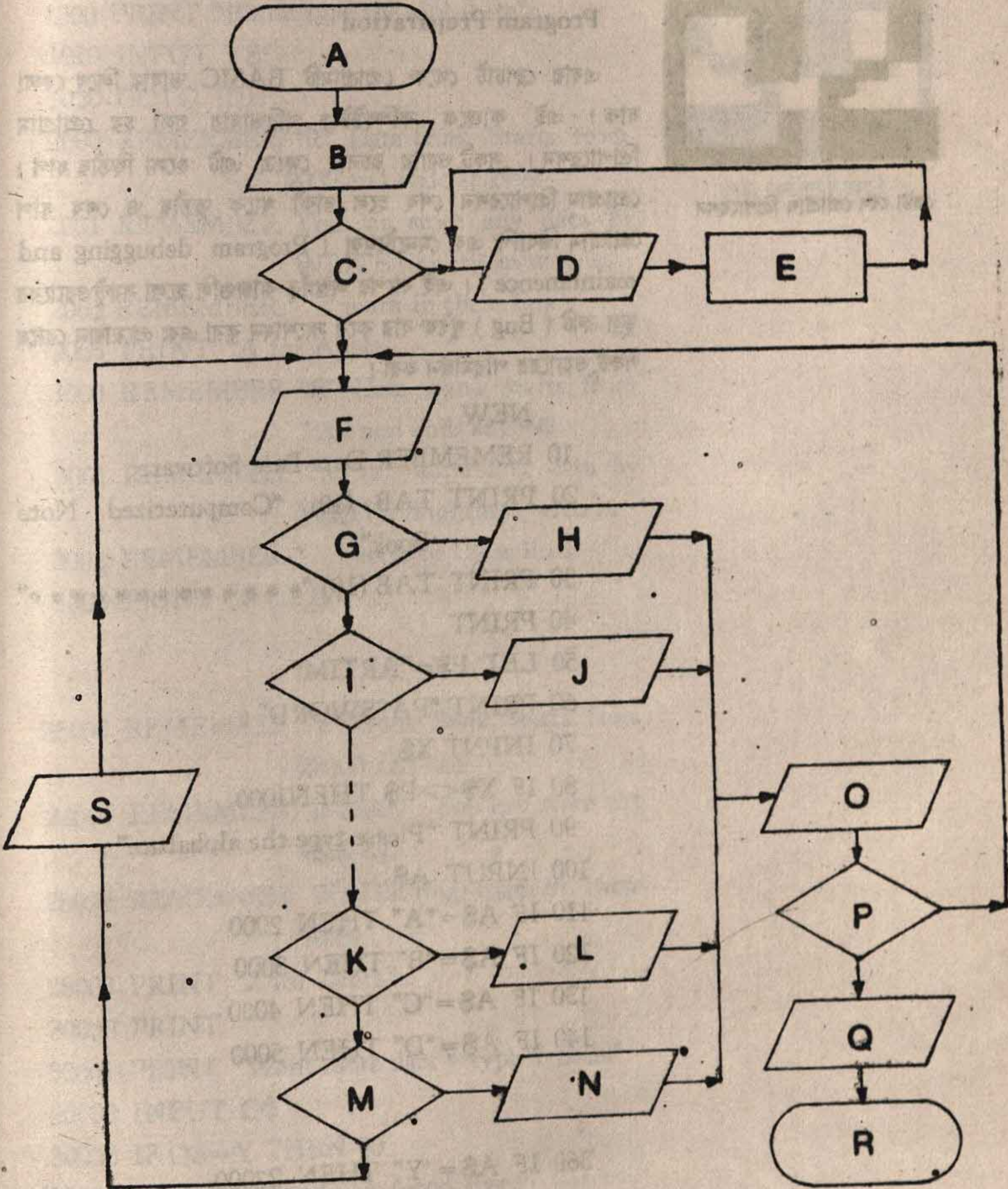
O = আরো তথ্য দরকার ?

P = যদি উত্তর হ্যাঁ হয় তবে F ধাপে ফিরে যাও

Q = যদি উত্তর না হয় তবে Q-তে যাও এবং "Thank you" কথাটি ছাপো

R = শেষ করো

S = যদি প্রোগ্রামার কোনো হরফের পরিবর্তে স্পেশাল ক্যারেকটার বা সংখ্যা টাইপ করেন তবে ছাপো "No Such Data Bank" এবং F ধাপে ফিরে যাও।


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S

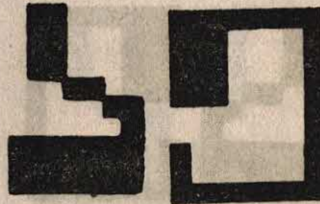
১০

ডেটা বেস প্রোগ্রাম প্রিপারেশন

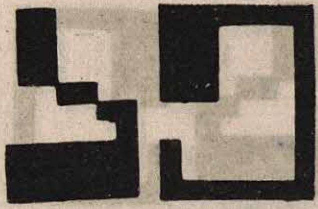
Program Preparation

এবার ফ্লোচার্ট থেকে প্রোগ্রামটি BASIC ভাষায় লিখে ফেলা যাক। এই কাজকে কম্পিউটার পরিভাষায় বলা হয় প্রোগ্রাম প্রিপারেশন। সফট্ওয়্যার রচনার ক্ষেত্রে এটি হলো দ্বিতীয় ধাপ। প্রোগ্রাম প্রিপারেশন শেষ হলে বাকী থাকে তৃতীয় ও শেষ ধাপ প্রোগ্রাম ডিবাগিং এবং মেনটেনেন্স (Program debugging and maintanence)। এই ধাপের অন্তর্গত কাজগুলি হলো সফট্ওয়্যারের ভুল ত্রুটি (Bug) খুঁজে বার করে সংশোধন করা এবং প্রয়োজন বোধে সফট্ওয়্যারের পরিমার্জন করা।

```
NEW
 10 REMEMBER Data Base Software
 20 PRINT TAB (10) "Computerized Note
          Book"
 30 PRINT TAB (10) "* * * * * * * * * * * * *"
 40 PRINT
 50 LET P$="ARTIM"
 60 PRINT "PASSWORD" ;
 70 INPNT X$
 80 IF X$<>P$ THEN1000
 90 PRINT "Please type the alphabate" ;
100 INPUT A$
110 IF A$="A" THEN 2000
120 IF A$="B" THEN 3000
130 IF A$="C" THEN 4000
140 IF A$="D" THEN 5000
 :
 :
360 IF A$='Y" THEN 27000
370 IF A$="Z" THEN 28000
380 PRINT "No such Data Bank"
390 GOTO 90
```




```
1000 PRINT "Please type the alphabate" ;
1010 INPUT Y$
1020 GOTO 1000
2000 REMEMBER "R" Data Bank  Starts  from
                  2000 and ends at 2999
2001 REMEMBER You can store  any  data  by
                  WRITE (ing) them with in
2002 REMEMBER "   " mark in these line
2003 PRINT "A for Apple               "
3000 REMEMBER "B"  Data  Bank  starts  from
                  3003 and ends at 3999
3001 REMEMBER You can  store  any  data  by
                  WRITE (ing) them with in
3002 REMEMBER "   " mark in these lines
3003 PRINT "B for Boy                 "
  :
  :
28000 REMEMBER "Z" Data  Bank  starts  from
                  28000 and end
28001 REMEMBER at 28999.  You can store any
                  data by
28002 REMEMBER  WRITE (ing) them in  these
                  lines
28003 PRINT "Z for Zebra"
30000 PRINT
30001 PRINT "Want more data ? type Y or N"
30002 INPUT Q$
30003 IF Q$=Y THEN 90
30004 PRINT "Thank you, Good Bye !"
30000 END
```



ডেটা বেস সফট্‌ওয়্যার

প্রোগ্রামটি RUN করার পর কম্পিউটার প্রথমে Computerized Note Book কথাটি ফুটিয়ে তুলবে। এরপর প্রোগ্রামারের কাছে সংকেত বাক্য বা Password জানতে চাইবে। সঠিক সঙ্কেত "ARTIM" কী বোর্ডে টাইপ করা হলে তবেই প্রোগ্রামের ডেটা ব্যাঙ্কে সঞ্চিত তথ্য পড়া যাবে। যদি ভুল সঙ্কেত বাক্য টাইপ করা হয়, তবে মনিটর স্ক্রিনে "Please type the alphabate" কথাটি এক নাগাড়ে ফুটে থাকবে। বলাই বাহুল্য, 50 নম্বর লাইনের পরিবর্তন ঘটিয়ে আপনি ইচ্ছেমত সঙ্কেত বাক্যটি পাল্‌টে নিতে পারবেন।

ধরা যাক প্রোগ্রামার সঠিক বাক্য টাইপ করলেন। তাহলে কম্পিউটার মনিটর স্ক্রিনে "Please type the alphabate" কথাটি ফুটিয়ে তুলে আপনার কাছে সত্যি সত্যিই প্রয়োজনীয় ডেটা ব্যাঙ্কের অদ্যাক্ষর জানতে চাইবে।

এবার আপনি যে হরফটি টাইপ করবেন, কম্পিউটার সেই ডেটা ব্যাঙ্কের অন্তর্গত যাবতীয় তথ্য মনিটর স্ক্রিনে লিখে দেবে। যদি আপনি A টাইপ করেন তবে স্ক্রিনে ফুটে উঠবে "A" ডেটা ব্যাঙ্কের অন্তর্গত সমস্ত তথ্য (এক্ষেত্রে সেটি হবে "A for Apple")। এই তথ্যগুলি কম্পিউটারে প্রোগ্রামটি টাইপ করার সময়ে প্রতিটি ডেটা ব্যাঙ্কের অব্যবহৃত স্টেটমেণ্ট লাইনগুলিতে PRINT কী ওয়ার্ডের সাহায্যে আপনাকে আগে থাকতে লিখে রাখতে হবে। যেমন 2003 লাইনে লেখা A for Apple কথাটি মুছে সেখানে যদি A অদ্যাক্ষরের কোনো দরকারী তথ্য (যেমন A দিয়ে নাম শুরু হয়েছে এমন কোনো বন্ধুর নাম ও ঠিকানা) লিখে রাখেন তবে সেটিই স্ক্রিনে ফুটে উঠবে। এইভাবে 2003 থেকে 2199, 3003 থেকে 3999 ইত্যাদি লাইনগুলিতে তথ্য সঞ্চয় করা যাবে।

তবে অন্যান্য Data Base সফট্‌ওয়্যারের থেকে এই সফট্‌-ওয়্যারটির প্রধান অসুবিধা হলো, এখানে তথ্যগুলি সবই মূল প্রোগ্রামের মধ্যে লিখতে হবে, কারণ Enhanced BASIC dialect ব্যবহার না করে কোন মতেই ডেটা ব্যাঙ্কের তথ্যকে সেকেণ্ডারী স্টোরেজ ব্যবস্থায় লেখা/পড়া যায় না। ফলে তথ্যগুলি কম্পিউটারের অমূল্য RAM স্মৃতি দখল করে নেবে। ছোট RAM স্মৃতি ভাণ্ডারযুক্ত কম্পিউটারে এই সফট্‌ওয়্যার তাই বিশেষ কাজে

লাগবে না। এটি ভালোভাবে ব্যবহার করতে হলে অন্তত 48K RAM যুক্ত বিশিষ্ট্য কম্পিউটার ব্যবহার করতে হবে।



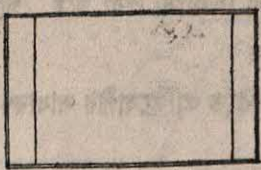
সাবরুটিন

কম্পিউটার পরিচালিত কুইজ প্রতিযোগিতা

এবার আরো বড় এবং জটিল একটি প্রোগ্রাম ফ্লোচার্টের সাহায্য নিয়ে তৈরী করে ফেলা যাক। তবে তার আগে আমাদের আর একটি ফ্লোচার্ট সঙ্কেতের সঙ্গে পরিচিত হতে হবে। সেটি হলো সাবরুটিন সিম্বল (Subroutine Symbol)।

Subroutine Symbol

মূল প্রোগ্রামের অন্তর্গত কোনো ছোট এবং স্বয়ং সম্পূর্ণ প্রোগ্রাম হলো সাবরুটিন, একথা আমরা আগের পরিচ্ছেদেই জেনেছি। মূল প্রোগ্রামের যে কোনো অংশ থেকে C.P.U. নিয়ন্ত্রণকে সাবরুটিনে নিয়ে যাওয়া যায়। সাবরুটিনের নির্দেশ পালন করার পর C.P.U নিয়ন্ত্রণ মূল প্রোগ্রামের যে অংশ থেকে সাবরুটিনে প্রবেশ করেছিল, ঠিক তার পরের অংশেই ফিরে আসে। এই সাবরুটিনকে যদি ফ্লোচার্টে পূর্বের ছটি সঙ্কেতের সাহায্যে আঁকা হয় তাহলে প্রোগ্রামের বিভিন্ন অংশ থেকে ফ্লো লাইন বারবার সাবরুটিনে যুক্ত করার ফলে ফ্লোচার্টটি দুর্বোধ্য হয়ে যায়। এই অসুবিধা এড়াতে বিশেষ সাবরুটিন সিম্বল্ ব্যবহার করা হয়।

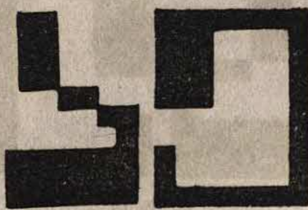


প্রোগ্রামিং অ্যানালিসিস

এবার কুইজ প্রতিযোগিতার অ্যানালিসিস শুরু করা যাক।

প্রতিযোগিতার নিয়মাবলী

কুইজ প্রতিযোগিতায় সাধারণতঃ দুটি প্রতিযোগী দল থাকেন। প্রতিযোগিতাটি পরিচালনা করেন প্রশ্নকর্তা বা কুইজ মাস্টার। উত্তর



কম্পিউটার পরিচালিত কুইজ কনটেস্ট

সঠিক হলে উত্তরদাতা তুপয়েন্ট পান আর ভুল হলে শূন্য। প্রতিপক্ষ উত্তর দিতে অসমর্থ হলে সেই উত্তর যদি অপর দল দিতে পারেন তবে সঠিক উত্তরদাতা এক বোনাস পয়েন্ট পান। প্রতিযোগীদের পয়েন্টের হিসেব রাখেন একজন স্কোরার।

তবে কম্পিউটারের মাধ্যমে কুইজ প্রতিযোগিতা পরিচালনার সময় প্রতিপক্ষ দল, কুইজ মাস্টার এবং স্কোরারের আদৌ প্রয়োজন হয় না। এই তিনটি কাজ কম্পিউটারকে দিয়ে সফট্‌ওয়্যারই করিয়ে নেয়। প্রতিযোগী একা কম্পিউটারের সাহায্যে তাঁর সাধারণ জ্ঞান যাচাই করে নিতে পারেন। এ ধরনের সফট্‌ওয়্যার স্কুল কলেজের ছাত্র ছাত্রীদের জন্য অত্যন্ত উপযোগী।

প্রবলেম ডেফিনেশন

(1) প্রোগ্রামের শুরুতে প্রতিযোগীর নাম জিজ্ঞাসা করা হবে

(2) প্রতিযোগীর নাম ধরে সম্ভাষণ করে প্রতিযোগীতার বিষয় নির্বাচন করতে আহ্বান জানানো হবে।

(3) প্রতিযোগিতার নিয়ম জানিয়ে দেওয়া হবে

(4) প্রশ্ন করা হবে

(5) উত্তর সঠিক হলে প্রতিযোগীকে অভিনন্দিত করা হবে।

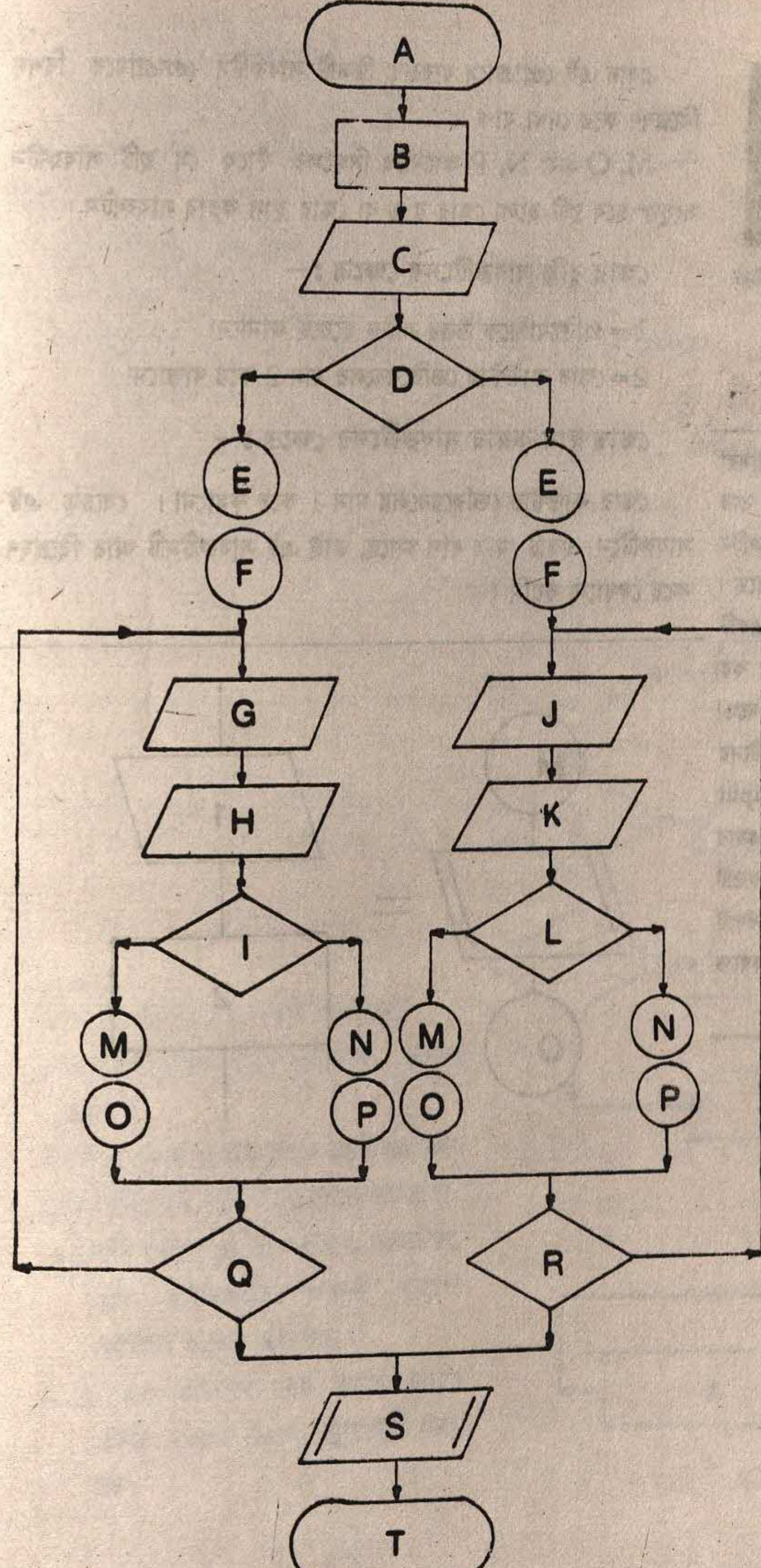
(6 উত্তর ভুল হলে সঠিক উত্তর বলে দেওয়া হবে এবং সান্ত্বনা দেওয়া হবে।

(7) সব কটি প্রশ্ন জিজ্ঞাসা করা হয়ে গেলে প্রতিযোগীর স্কোর জানানো হবে।

(8) স্কোরের ভিত্তিতে প্রতিযোগীর সাধারণ জ্ঞানের মান জানানো হবে।

ফ্লোচার্ট নির্মাণ

ওপরে লিখে দেওয়া প্রবলেম ডেফিনেশন অনুসারে ফ্লোচার্টটি এঁকে ফেলা যাক। এই প্রোগ্রামটি বাহ্যত সহজ মনে হলেও ফ্লোচার্টে চোখ বোলালেই এর জটিলতা উপলব্ধি করতে পারবেন।



A = প্রোগ্রামের শুরু

B = স্কোর (Score) গণনাকারী ভেরিয়েবলের প্রাথমিক মান নির্দিষ্টকরণ (initialization)

C = প্রতিযোগীতার বিষয় নির্বাচন করতে অনুরোধ জানানো

D = যদি বিষয়টি বিজ্ঞান হয় তবে বিজ্ঞান বিষয়ক ব্রাঞ্চে প্রোগ্রামকে নিয়ে যাওয়া। অথবা যদি বিষয়টি সাহিত্য ও কলা বিষয়ক হয় তবে ওই বিষয়ক ব্রাঞ্চে নিয়ে যাওয়া।

E, F = প্রতিযোগীতার নিয়ম জানানোর সাবরুটিনে যাওয়ার Connector Symbol

G, J = বিজ্ঞান বিষয়ক প্রশ্ন হলে G ব্রাঞ্চ এবং সাহিত্য কলা বিষয়ক প্রশ্ন হলে J ব্রাঞ্চে যেতে হবে। G বা J-তে রক্ষিত আছে প্রশ্নপত্রের প্রশ্নগুলি। এখানে ফ্লোলাইন এলে প্রথম প্রশ্ন জিজ্ঞাসা করা হবে

H, K = প্রশ্নের উত্তর পড়া হবে।

I, L = উত্তর সঠিক হলে M, O-তে অথবা বেঠিক হলে N, P যে নিয়ন্ত্রণ যাবে।

M, O = স্কোর বৃদ্ধি করা সাবরুটিন সংযুক্ত হবে M, O-তে

N, P = স্কোর হ্রাস করার সাবরুটিন সংযুক্ত হবে N, P-তে

Q, R = সব কটি প্রশ্ন করা শেষ না হলে পুনরায় G, J ধাপে ফিরে যেতে হবে আর যদি হয়ে গিয়ে থাকে তবে S সাবরুটিনে যেতে হবে

S = মোট স্কোর জানানোর সাবরুটিন

T = প্রোগ্রাম শেষ।

এবার এই প্রোগ্রামে ব্যবহৃত তিনটি সাবরুটিন প্রোগ্রামকে বিশদ বিশ্লেষণ করে দেখা যাক।

M, O এবং N, P কানেক্টর সিম্বলের ফাঁকে যে দুটি সাবরুটিন সংযুক্ত হবে দুটি হলো স্কোর বৃদ্ধি বা স্কোর হ্রাস করার সাবরুটিন।

স্কোর বৃদ্ধি সাবরুটিনের ক্ষেত্রে :—

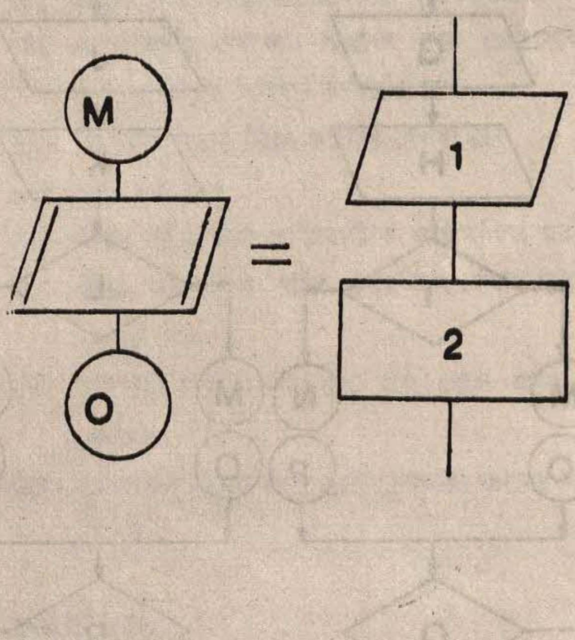
1 = প্রতিযোগীকে উত্তর সঠিক হয়েছে জানানো

2 = স্কোর কাউন্টার ভেরিয়েবলের মান 2 করে বাড়ানো

স্কোর হ্রাস করার সাবরুটিনের ক্ষেত্রে :—

স্কোর কাউন্টার ভেরিয়েবলের মান 1 করে কমানো। যেহেতু এই সাবরুটিনে একটি মাত্র ধাপ আছে, তাই এই সাবরুটিনটি আর বিশ্লেষণ করে দেখানো হয়নি।

যে সাবরুটিন সিম্বলের সঙ্গে আমরা একটু আগে পরিচিত হয়েছি, তার সঙ্গে কিন্তু এখানে ব্যবহৃত সাবরুটিন সংকেতটির সামান্য পার্থক্য আছে। ANSI-এর সংকেতটি হল একটি Rectangle, আর এখানে ব্যবহার করা হয়েছে Input/output সংকেতের মতো parallelogram. আসলে সাবরুটিনের অন্তর্গত এক বা একাধিক Output Symbol-এর উপস্থিতি জানান দেওয়ার জন্যই এভাবে সাবরুটিন সংকেতটি আঁকা হয়েছে। তবে কার্যক্ষেত্রে সর্বদাই ANSI-এর সংকেতটিই ব্যবহার করতে হবে।



সাবরুটিন S-এর বিশ্লেষণ :—

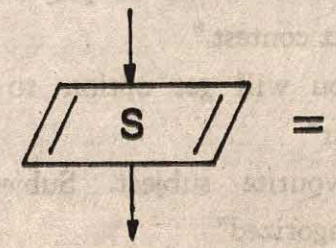
মোট স্কোর জানান দেওয়ার সাবরুটিন হল S, এর কাজ কেবল স্কোর জানান দেওয়াই নয়, স্কোরের ভিত্তিতে প্রতিযোগীর মান বিশ্লেষণ করা।

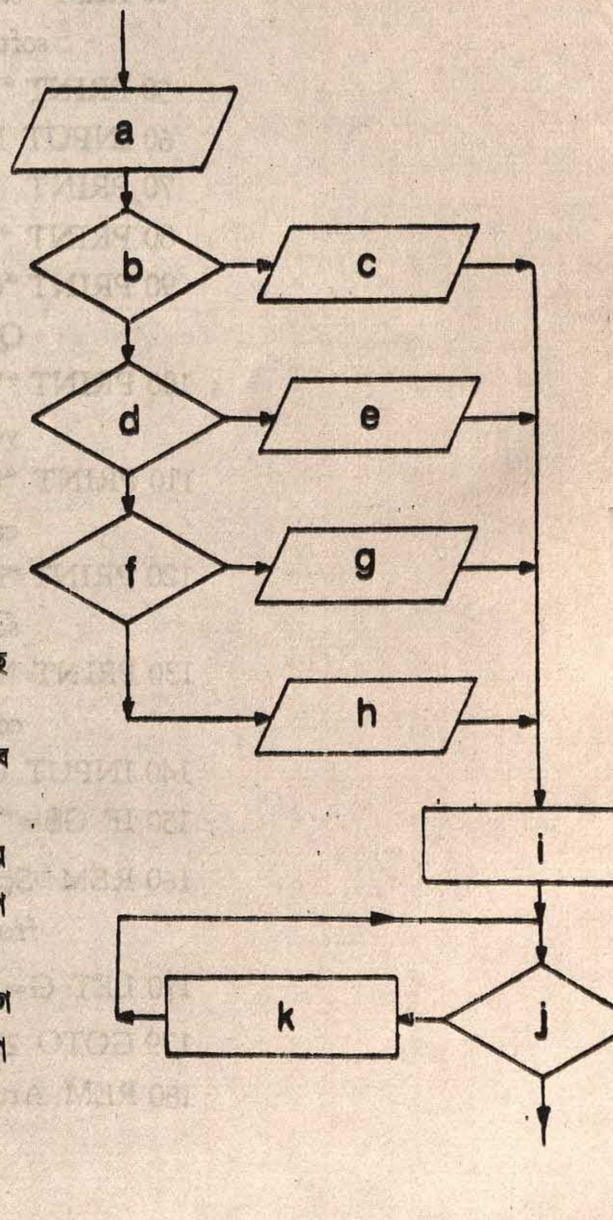
কম্পিউটার পরিচালিত কুইজ কনটেস্ট

a = মোট স্কোর জানানো

b = স্কোর কি 10-এর মধ্যে 8 অথবা 10 ? যদি তাই হয় তবে

c = প্রতিযোগীকে সব থেকে ভাল ফল করার জন্য অভিনন্দন জানানো





d = যদি স্কোর 5-এর মধ্যে হয় তবে

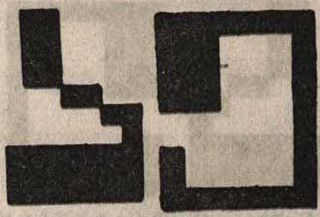
e = প্রতিযোগীর স্কোর সাধারণ হয়েছে জানানো

f = যদি স্কোর শূন্যের থেকে বেশী হয় তবে

g = প্রতিযোগী পাস করে গেছে জানানো

h = যদি স্কোর শূন্য বা ঋণাত্মক সংখ্যা হয় তবে প্রতিযোগী অত্যন্ত খারাপ ফলাফল করেছে জানানো।

i, j, k = প্রোগ্রাম শেষ করার আগে একটি time delay লুপ সৃষ্টি করা হয়।



কম্পিউটার নিয়ন্ত্রিত কুইজ কনটেস্ট।

এবার ফ্লোচার্ট থেকে BASIC-এ প্রোগ্রামটি লিখে ফেলা যাক :—

```
  5 LET L=O
 10 REM Computerized Quiz contest demo
        program
 20 REM
 30 REM Can also be used as computerized
        interview
 40 REM or Computerized teaching aid
        software
 50 PRINT "Hello, what is your name" ;
 60 INPUT N$
 70 PRINT
 80 PRINT "I am very glad to meet you" ; N$
 90 PRINT "Come on, let us play a game of
          Quiz contest."
100 PRINT "You will get option to choose
          your"
110 PRINT "favourite subject. Subjects are
          catagorized"
120 PRINT "into two main groups, arts &
          science."
130 PRINT "In which group you want to
          compete" ;
140 INPUT G$
150 IF G$="Arts" THEN |190|
160 REM Science question bank starts
        from |3000|
170 LET G= |3000|
179 GOTO 210
180 REM Arts question bank starts from |2000|
```

```
190 LET G= 2000
200 REM Rules & regulation display subrouting
        starts from 1000
210 GOSUB 1000
220 GOSUB G
230 REM Score display subrouting starts
        from 6000
240 GOSUB 6000
250 REM
260 REM Here only five questions are stored in
270 REM each question bank. You can
        always
280 REM increase the number of questions or
        even change them by
290 REM modifying the software
300 REM in between the
310 REM line numbers 2000 to 2999 and
320 REM 3000 to 3999
330 STOP
1000 PRINT "Rules and regulation of the Quiz
        Contest"
     PRINT "*******************"
1010 PRINT
1020 PRINT "Each question will be accom-
        panied by
1030 PRINT "several probable answers. You
        only have"
1040 PRINT "to type back the correct answer
        number"
```

```
1050 PRINT "For each correct answer you will
         be rewarded"
1060 PRINT with 2 points. But for each
         wrong answer"
1070 PRINT "1 point will be deducted from
         your"
1080 PRINT "Score. Remember it is compulsory
         to answer"
1090 PRINT "every question"
1100 PRINT
1110 PRINT "Are you ready" ; N$ ; "? Type
Yes or No"
1020 INPUT A$
1030 IF A$="NO" THEN 1000
1040 RETURN
2000 PRINT "Who pointed the famous fresco
         the last Supper ?"
2001 PRINT "1. Botechelli 2. Michel Angelo
         3. Leonardo Da Vinci 4. Gaugin."
2002 INPUT A
2003 IF A=3 THEN 2007
2004 PRINT "Wrong. The correct answer is
         Leonardo Da Vinci"
2005 GOSUB 7000
2006 GOTO 2008
2007 GOSUB 8000
2008 PRINT "Which epic is the greatest epic of
         the world ?"
2009 PRINT "1. Iliad 2. Ramayana 3. Oddyssey
         4. Mahabharatha"
2010 INPNT B
2011 IF B=4 THEN 2016
```

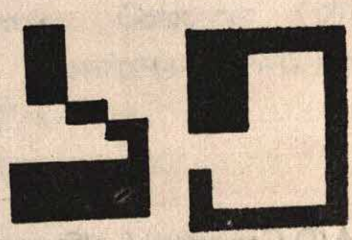
```
2012 PRINT "Sorry"; N$; "wrong answer.
          The correct"
2013 PRINT "answer is Mahabharatha."
2014 GOSUB 7000
2015 GOTO 2017
2016 GOSUB 8000
2017 PRINT "Pieta is a world famous—"
2018 PRINT "(1) Sculpture (2) Novel (3) Paint-
          ing (4) Architecture"
2019 INPUT C
2020 IF C=1 THEN 2024
2021 PRINT "Wrong answer. The correct
          aneswer is 1"
2022 GOSUB 7000
2023 GOTO 2025
2024 GOSUB 8000
2025 PRINT "Sonnet is kind of"
2026 PRINT (1) Song (2) Staire (3) Poem
          (4) Musical instrument."
2027 INPUT D
2028 IF D=3 THEN 2032
2029 PRINT "Beter luck next time, the correct
          answer in 3."
2030 GOSUB 7000
2031 GOTO 2033
2032 GOSUB 8000
2033 PRINT "Picolo is kind of"
2034 PRINT "(1) Paint (2) World famous
          Novel (3) Pluck string instrument
          (4) Small flute."
```

```
2035 INPUTE
2036 IF E=4 THEN 2040
2037 PRINT "Bad luck" ; T$ ; "The correct
          answer is 4
2038 GOSUB 7000
2039 GOTO 2041
2040 GOSUB 8000
2041 PRINT "Game is over !"
2042 RETURN
3000 PRINT "Who invented Diode valve ?"
3001 PRINT (1) Ted Hoff (2) Lee De Forest
          (3) Edison (4) Faraday"
3002 INPUT F
3003 IF F=(2) THEN 3007
3004 PRINT "Wrong answer. The correct
          answer is 2."
3005 GOSUB 7000
3006 GOTO 3008
3007 GOSUB 8000
3008 PRINT "Alfred Nobel was the invention
          of"
3009 PRINT "(1) Life saving drugs (2) Dyna-
          mite (3) Submarine (4) Atom
          bomb."
3010 INPUT H
3011 IF H=2 THEN 3015
3010 PRINT "No, the correct answer is 2
3013 GOSUB 7000
3014 GOTO 3016
3015 GOSUB 8000
3016 PRINT "Essential element which must
          exist in a organic compound is"
```

```
3017 PRINT "(1) Oxygen (2) Nitrogen (3) Sul-
          phur (4) Carbon (5) Hydrogen
3018 INPUT I
3019 IF I=4 THEN 3023
3020 PRINT "Sorry" ; N$ ; "the correct answer
          is 4."
3021 GOSUB 7000
3022 GOTO 3024
3023 GOSUB 8000
3024 PRINT "The term closely related with
          Hall effect is"
3025 PRINT "(1) Magnetic fiele (2) Static
          charge (3) Plazma (4) Radio
          activity
3026 INPUT J
3027 IF J=1 THEN 3031
3028 PRINT "Incorrect, the correct answer is 1"
3029 GOSUB 7000
3030 GOTO 3032
3031 GOSUB 8000
3032 PRINT "Eienstien received Nobel prize
          for"
3033 PRINT "(1) Theory of Relativity (2) In-
          vention of atom bomb (3) Rese-
          arch on Photo electricty (4) Bose
          Eienstien Statistics."
3034 INPUT K
3035 IF K=3 THEN 3039
3036 PRINT "Hard luck. The correct answer
          is 3."
```

```
3037 GOSUB 7000
3038 GOTO 3040
3039 GOSUB 8000
3040 PRINT "The Game is over !"
3041 RETURN
6000 PRINT
6001 PRINT "Well" ; N$ ; "Your score is" ; L
6002 IF L≥8 THEN 6007
6003 IF L=5 THEN 6008
6004 IF L>0 THEN 6009
6006 GOTO 6010
6007 PRINT "I must admit you are very strong
              in this subject."
6008 PRINT "You have done well."
6009 PRINT "You are passed, but I think you
              must update your knowledge in
              this subject."
6010 PRINT "I am sorry to say that you know
              nothing about this subject."
6015 REM Time delay loop for easy reading
6020 FOR X=1 TO 1000
6030 NEXT X
6040 RETURN
7000 REM Score deductor subroutione
7001 LET L=L-1
7002 RETURN
8000 REM Score increaser subroutine
8001 PRINT "Correct !"
8002 LET L=L+2
8003 RETURN
10000 END
```

প্রোগ্রামটিতে একাধিক ব্রাঞ্চ থাকার দরুন প্রোগ্রামটি মোটামুটি সাড়ে চোদ্দো হাজার (14500) রকম ভাবে বিভিন্ন ব্রাঞ্চে ঘুরতে পারে ! তাই এটি RUN করে বইয়ের পাতায় দেখানোর থেকে কোনো কম্পিউটারে Entry করে চালিয়ে দেখুন, তাহলেই User friendly এই সফট্‌ওয়্যারটির পুরো মজা পাবেন।



শুভ প্রোগ্রামিং !

শেষ করার আগে একটা কথা বলে রাখি, সেটি হলো যে কোনো সমস্যা সমাধানের সফট্‌ওয়্যারেই বিভিন্ন লজিকাল পাথ (Logical path) ধরে অগ্রসর হওয়া যায় (যেমন এসপ্ল্যানেড থেকে এয়ারপোর্ট যেতে আমরা যে কোনো রাস্তা বেছে নিতে পারি)। কিন্তু প্রোগ্রামারকে সর্বদা খেয়াল রাখতে হয় যাতে তাঁর বেছে নেওয়া রাস্তাটাই সংক্ষিপ্ততম হয়। এর ফলে কম্পিউটারের আভ্যন্তরীণ RAM ভাণ্ডার প্রোগ্রামের দ্বারা সম্পৃক্ত (saturated) হয়ে যাবার সম্ভবনা কম হবে। সেই সঙ্গে এটাও বলে রাখি, এই বইয়ে দেওয়া সমাস্ত প্রোগ্রাম গুলি আমি যেভাবে ফ্লোচার্ট এঁকে সমাধান করেছি, সেটিই একমাত্র সমাধানের পথ নয়। আপনি ইচ্ছা করলে ওই সফট্‌ওয়্যারগুলির মধ্যে কয়েকটিকে আরো সংক্ষেপে ছোটো করে লিখতে পারবেন। বিশেষ করে কম্পিউটার-নিয়ন্ত্রিত কুইজ প্রতিযোগিতা সফট্‌ওয়্যারটি FOR/NEXT এবং READ/DATA কী ওয়ার্ড ব্যবহার করে আরো সংক্ষিপ্ত করা সম্ভব। তবে এটা কিন্তু আমি বলবো না, আপনাকেই মাথা খাটিয়ে লিখে ফেলতে হবে।

শুভ প্রোগ্রামিং !

পরিশিষ্ট

কম্পিউটার অভিধান

কম্পিউটার প্রযুক্তি সম্বন্ধে পড়াশোনা করতে গেলে আমাদের হামেশাই বহু নতুন টেকনিক্যাল শব্দের সম্মুখীন হতে হয়। এই পরিচ্ছেদে বহু ব্যবহৃত সেই সব ইংরেজী শব্দগুলির অর্থ বাংলায় সহজ করে ব্যাখ্যা করা হয়েছে। মূল শব্দগুলি যেহেতু ইংরেজীতেই লেখা হয় তাই এখানেও শব্দগুলি ইংরিজিতেই লেখা হয়েছে এবং খুঁজে পাবার সুবিধার জন্য ইংরেজি বর্ণমালার ক্রম অনুসারে সাজানো হয়েছে।

A

Abacus-প্রাচীনতম গণনা সহায়ক যন্ত্র।

Access—স্মৃতি ভাণ্ডারে নূতন তথ্য লেখা বা লিখিত তথ্য খুঁজে বার করার জন্য মেমারী সেল গুলির সঙ্গে সংযোগ স্থাপন করা।

Access time—'স্মৃতি ভাণ্ডার থেকে তথ্য পড়ো' নির্দেশ দেওয়ার মুহূর্ত থেকে, তথ্য খুঁজে পড়ে ফেলা সম্পূর্ণ হওয়া পর্যন্ত যে সময় লাগে।

Acoustic Coupler—বিশেষ ধরনের মডেম (Modem), যার সাহায্যে সাধারণ টেলিফোন রিসিভারের মাধ্যমে দুটি বিভিন্ন জায়গায় অবস্থিত কম্পিউটারের মধ্যে সংযোগ স্থাপন করা যায়।

Accumulator—সেন্ট্রাল প্রসেসিং ইউনিটের অভ্যন্তরস্থ অস্থায়ী RAM স্মৃতি।

Adder—বাইনারি পদ্ধতিতে যোগ করার ইলেক্ট্রনিক সার্কিট।

Address—কোন বিশেষ মেমরি সেলের পরিচয় জ্ঞাপক নম্বর।

Algol—Algorithmatic Language কথাটির সংক্ষিপ্ত রূপ। এটি ইঞ্জিনিয়ারিং সমস্যা সমাধানের জন্য বিশেষভাবে তৈরী হাই লেভেল প্রোগ্রামিং ভাষা। ইউরোপে প্রচলিত এবং জনপ্রিয়তার দিক থেকে এর স্থান FORTRAN এর পরে।

Algorithm—কম্পিউটারের সাহায্যে কোনো সমস্যা সমাধানের যুক্তি গ্রাহ্য পথ প্রদর্শনকারী তালিকা (Flow Chart দেখুন)।

Alphameric—Alphanumeric দেখুন।

Alphanumeric— হরফ এবং সংখ্যা।

ALU—Arithmatic Logic Unit কথার সংক্ষিপ্ত রূপ। এটি একটি ইলেক্ট্রনিক সার্কিট এবং এর সাহায্যে কম্পিউটারে যাবতীয় গাণিতিক সমস্যা সমাধান করা হয়।

Analog Computer—কম্পিউটারের একটি বিশেষ প্রজন্ম। এই কম্পিউটার Analog সংকেতের মাধ্যমে কাজ করে।

Annotation—ফ্লো চার্টের বিভিন্ন ধাপকে ব্যাখ্যা করার জন্য ব্যবহৃত চিহ্ন বা সংকেত।

APL—A Programming Language কথাটির সংক্ষিপ্ত রূপ। IBM কোম্পানী এই হাই লেভেল ল্যাংগোয়েজটি তৈরী করেছেন। প্রচুর লাইব্রেরী ফাংশন এবং user defined instruction এর সুবিধার জন্য APL একটি অত্যন্ত শক্তিশালী প্রোগ্রামিং ভাষা। একমাত্র বড় মেইন ফ্রেম কম্পিউটারেই এই ভাষা ব্যবহার করা হয়।

Array—একই ধরনের ভেরিয়েবলের ম্যাট্রিক্স হিসেবে সাজানো সারণী। এই সারণীতে সাজানো ভেরিয়েবল গুলি সব একই ইংরেজী হরফ দিয়ে লেখা হয়। প্রতিটি ভেরিয়েবলকে আলাদা করে চেনার জন্য হরফের সঙ্গে বিভিন্ন সাফিক্স (Suffixe) ব্যবহার করা হয়।

Assembly language—কম্পিউটার প্রোগ্রামিং এর লো লেভেল ভাষা। এই ভাষা মেশিন ল্যাংগোয়েজ এবং হাইলেভেল ল্যাংগোয়েজের অন্তর্বর্তী বলা চলে।

Assignment—হাই লেভেল কম্পিউটার ভাষায়, যে স্টেটমেন্টের সাহায্যে কোনো ভেরিয়েবলের একটি নির্দিষ্ট মান ঠিক করে দেওয়া হয়, তাকে অ্যাসাইনমেন্ট স্টেটমেন্ট বলে। উদাহরণঃ BASIC ভাষার let স্টেটমেন্ট।

B

Backing Storage—বাড়তি স্মৃতি ভাণ্ডার, যেমন হার্ড ডিস্ক, ফ্লপি, ম্যাগনেটিক টেপ ইত্যাদি।

Bar Code—কালো সাদা দাগের সাহায্যে লিখিত সংকেত। বার কোড রিডার যন্ত্র এই সংকেত পড়ে তার অন্তর্নিহিত অর্থ কম্পিউটারে পাঠায়।

Base— Radix দেখুন

BASIC—Beginners All-purpose Symbolic Instruction Code কথাটির সংক্ষিপ্ত রূপ। সর্বাধিক জনপ্রিয় মাইক্রো কম্পিউটার প্রোগ্রামিং হাই লেভেল ল্যাংগোয়েজ।

Batch processing—কম্পিউটার প্রোগ্রামিংএর একটি বিশেষ পদ্ধতি। এই পদ্ধতিতে একই ধরনের তথ্য, যেমন বাণিজ্যিক লেনদেন প্রভৃতি, এক সঙ্গে প্রসেস করা হয়।

Baud—ডিজিটাল তথ্য প্রেরণ গতিবেগের একক।

BCD—Binary Coded Decimal কথাটির সংক্ষিপ্ত রূপ। দশমিক সংখ্যাকে বাইনারি সংখ্যায় প্রকাশ করার পদ্ধতি।

Binary—একটি বিশেষ সংখ্যা পদ্ধতি, যাতে মাত্র দুটি সংখ্যা 1 এবং 0 র মাধ্যমে সমস্ত সংখ্যা প্রকাশ করা হয়। ডিজিটাল কম্পিউটার এই সংখ্যা পদ্ধতির সাহায্যেই গণনা কার্য সম্পন্ন করে। বাইনারি পদ্ধতির র‍্যাডিক্স হলো 2

Bit—বাইনারি সংখ্যা পদ্ধতির 1 এবং 0 র মধ্যে যে কোন একটি সংখ্যা।

Bit rate—কম্পিউটার প্রতি সেকেণ্ডে কতো বিট (Bit) তথ্য আদান প্রদান করতে পারে তারই পরিচয় জ্ঞাপক সংখ্যা।

Block diagram (১) ফ্লো চার্টেরই একটি বিশেষ রূপ, যেখানে কেবল মাত্র গুরুত্বপূর্ণ নির্দেশ গুলিই প্রদর্শন করা থাকে।

(২) কোন জটিল সার্কিট বা যন্ত্রকে অতি সরলিকৃত করে ছবির মাধ্যমে ব্যাখ্যা করা।

Buffer—ছোট এবং মাঝারি অস্থায়ী RAM স্মৃতি ভাণ্ডার। কম্পিউটার প্রিন্টারে ব্যবহার করা হয়। কম্পিউটার তার ফলাফল যত দ্রুত বলে দেয়, প্রিন্টার সমান তালে সেই তথ্য ছাপতে পারে না। সুতরাং বাড়তি তথ্য যেটি তখনও ছাপা যায়নি, সেটি প্রিন্টারের বাফারে জমা হতে থাকে। কম্পিউটারের আউটপুটে তথ্য আসা যখনই মুহূর্তের জন্য বন্ধ হয়, প্রিন্টার সেই অবকাশে বাফারে সঞ্চিত তথ্য গুলি ছেপে ফেলে।

Bug—সফট্ওয়্যারের কোনো ভুল ভ্রান্তি।

Bus—কম্পিউটার সার্কিটের তথ্য এবং নির্দেশ পরিবহনকারী বিদ্যুৎ পরিবাহী পথ।

Byte—আট বিট্ (bit) বাইনারি তথ্য মিলিয়ে এক বাইট (byte) তথ্য তৈরী হয়। কম্পিউটার স্মৃতি ভাণ্ডারের ধারন ক্ষমতা এই বাইট এককেই প্রকাশ করা হয়।

C

CAD/CAM—(Computer Aided Design /Computer Aided Manufacturing) কম্পিউটারের সাহায্যে যন্ত্রপাতির ডিজাইন করা এবং কলকারখানার যন্ত্রপাতি নিয়ন্ত্রণ করার পদ্ধতি।

Canned program—ফ্লপি ডিস্ক এবং ক্যাসেটে রেকর্ড করা কম্পিউটার প্রোগ্রাম। বর্তমানে বহু সফ্ট্ওয়্যার নির্মাতা বিভিন্ন কাজের জন্য তৈরী করা হাজার হাজার এ ধরনের প্রোগ্রাম বিক্রি করছেন।

Capacitor—বৈদ্যুতিক স্ট্যাটিক চার্জ জমিয়ে রাখার ব্যবস্থা। অনেক সময় কণ্ডেনসারও (condenser) বলা হয়।

Card—$7\frac{3}{8} \times 3\frac{1}{4}$ আকৃতির শক্ত বোর্ড কাগজে এই কার্ড তৈরী হয়। কার্ডে বারোটি সারি এবং আশিটি স্তম্ভের বিভিন্ন জায়গায় ফুটো করে কম্পিউটার প্রোগ্রামকে সাংকেতিক ভাবে লিপি বদ্ধ করা হয়। মেইন ফ্রেম কম্পিউটারের ক্ষেত্রে এটি হলো একটি অন্যতম প্রধান ইনপুট ডিভাইস।

Card hopper—কার্ড রিডার যন্ত্রে যে জায়গায় পাঞ্চ কার্ড রাখা হয়।

Card punch—কার্ড ফুটো করার যন্ত্র।

Card reader—পাঞ্চ কার্ডের ফুটো সংকেত থেকে তথ্য পড়ে, সেই তথ্যকে কম্পিউটার বোধ্য ডিজিটাল সিগন্যালে রূপান্তরিত করার যন্ত্র।

Cassette—ফিলিপস্ কোম্পানীর ডিজাইন করা ছোট প্লাস্টিক খাপে ভর্তি চৌম্বক ফিতে। খাপে ভর্তি থাকার দরুণ ওপেন রিল চৌম্বক ফিতের মতো খুলে যাওয়ার ভয় নেই। সাধারণ গান বাজাবার টেপ রেকর্ডারে ব্যবহার করা হয়।

Chad—কার্ড এবং টেপ পাঞ্চ করলে যে কুঁচি কাগজের টুকরো পাওয়া যায়।

Character—কম্পিউটার বোধ্য সমস্ত হরফ, গাণিতিক সংখ্যা এবং বিশেষ চিহ্ন। সাধারণত ইংরেজী হরফ A থেকে Z, সংখ্যা 0 থেকে 9, ফুলস্টপ, কমা, সেমিকোলোন, ড্যাশ, এক্সক্লামেশন, ইনভার্টেড কমা ইত্যাদি ধরনের চিহ্ন এবং গ্রাফিক্সে ব্যবহৃত বিন্দু, সরল রেখা, নানা ধরনের প্যাটার্ন ইত্যাদিকেই ক্যারেকটার বলা হয়।

Character Set—কম্পিউটারে ব্যবহৃত সমস্ত ধরনের ক্যারেকটার। প্রতিটি কম্পিউটারেরই নিজস্ব আলাদা ক্যারেকটার সেট থাকে।

Chip—সিলিকনের অত্যন্ত পাতলা এবং ছোট আয়তক্ষেত্রাকার টুকরো, যার ওপর IC, LSI এবং VLSI পদ্ধতিতে অতিক্ষুদ্রায়িত সার্কিট তৈরী হয়। বর্তমানে সমস্ত IC. LSI এবং VLSI কেই চিপ্ বলে অভিহিত করা হয়।

Circuit—ইলেকট্রিকাল এবং ইলেক্ট্রনিক যন্ত্রাংশ জুড়ে কোনো বিশেষ কাজের জন্য তৈরী বৈদ্যুতিক বর্তনী।

Clock—কম্পিউটার অভ্যন্তরস্থ বিশেষ ইলেক্ট্রনিক সার্কিট। এটি প্রতিনিয়ত বিদ্যুৎ স্পন্দন পাঠিয়ে C. P. U-এর তথ্য আদান প্রদানের হার নিয়ন্ত্রণ করে কম্পিউটারের clock pulse যত দ্রুত হয়, সেই কম্পিউটার তত দ্রুত গণনা করতে পারে।

CMOS—Complementary Metal Oxide Semiconductor কথাটির সংক্ষিপ্ত রূপ। IC তৈরী করার বিশেষ প্রযুক্তি।

COBOL—কমন বিজনেস ওরিয়েনটেড ল্যাংগোয়েজ (Common Business Oriented Language) কথাটিকে সংক্ষিপ্ত করে COBOL কথাটির উৎপত্তি। এটি ব্যবসা সংক্রান্ত সমস্যা সমাধান করার জন্য বিশেষ ভাবে লিখিত হাই লেভেল কম্পিউটার প্রোগ্রামিং ভাষা। এই ভাষায় সংকেতের থেকে সাধারণ ইংরেজী শব্দের ব্যবহার বেশী, তাই COBOL প্রোগ্রাম পড়ে, যাঁরা কম্পিউটার প্রোগ্রামিং আদৌ জানেন না, তাঁরাও অর্থ বুঝতে পারেন।

Coding—মেশিন ল্যাংগোয়েজে প্রোগ্রাম লেখার কাজ।

Coding Sheet—মেশিন ল্যাংগোয়েজে প্রোগ্রাম লেখার সুবিধার জন্য বিশেষ ছাপানো ফর্ম।

Core—ফ্যারাইটের (এক ধরনের চৌম্বক পদার্থ) তৈরী পুঁতি (bead)।

Core Storage—ফ্যারাইট পুঁতির চৌম্বক ক্ষেত্র পরিবর্তন করে ডিজিটাল তথ্য জমিয়ে রাখার ব্যবস্থা

Compiler—ROM কিম্বা EPROM ধরনের স্থায়ী স্মৃতিতে লিখিত বিশেষ সফট্ ওয়্যার প্রোগ্রাম, যার সাহায্যে CPU, হাই লেভেল ল্যাংগোয়েজে লিখিত প্রোগ্রামকে সঠিক ক্রমে মেশিন ল্যাংগোয়েজে অনুবাদ করতে পারে।

Computer Aided Design (cad)—কম্পিউটারের সাহায্য নিয়ে যন্ত্রপাতির নক্সা তৈরী করা।

Computer—একটি ইলেক্ট্রনিক যন্ত্র, যাতে নির্দেশ এবং তথ্য ভরে দিলে সেটি কোন বিশেষ সমস্যার সমাধান করে ফলাফল জানিয়ে দেয়।

Conductor—বিদ্যুৎ পরিবাহী।

Console—কী বোর্ড এবং মনিটর। এর সাহায্যে প্রোগ্রামার এবং কম্পিউটার পরস্পর যোগাযোগ স্থাপন করতে পারে।

Constant—প্রোগ্রামে লিখিত কোনো সংখ্যা বা হরফ, যার মান, কম্পিউটারের সাহায্যে সেই

প্রোগ্রাম সমাধান করা কালীন অপরিবর্তিত থাকে।

Counter—বৈদ্যুতিক স্পন্দন গোনার ইলেক্ট্রনিক সার্কিট।

CPS—Cycles Per Second—কথাটির সংক্ষিপ্ত রূপ।

Crash—ভুল প্রোগ্রাম অথবা সার্কিটের গোলযোগে কম্পিউটারের কাজ বন্ধ হয়ে যাওয়া।

Cursor—VDU স্ক্রিনের যে জায়গায় কম্পিউটার নতুন ক্যারেকটার ফুটিয়ে তুলবে, সেই স্থান নির্দেশকারী ক্রমাগত জ্বলতে নিভতে থাকা ছোট আয়তক্ষেত্রাকার আলোক চিহ্ন।

D

Daisy wheel—ডেইজি হুইল প্রিন্টারের ব্যবহৃত পরিবর্তনশীল ছাপার টাইপ।

Data—কম্পিউটার গণনা কার্যের জন্য প্রয়োজনীয় তথ্য। এটি প্রোগ্রামিং এর সময় কম্পিউটারে ঢুকিয়ে দেওয়া হয় অথবা গণনা কার্য চলাকালীন জোগান দেওয়া হয়।

Data Processing—কম্পিউটারের সাহায্যে তথ্য গ্রহণ এবং বিশ্লেষণ করে ফলাফল বলে দেওয়ার পদ্ধতি।

Debug—কম্পিউটার প্রোগ্রামের ত্রুটি সনাক্তকরণ এবং সংশোধন।

Decimal—দশমিক সংখ্যা পদ্ধতি। এই পদ্ধতিতে দশটি মূল সংখ্যা আছে (0 থেকে 9 অবধি)।

Dialect—প্রতিটি কম্পিউটার নির্মাতার নিজস্ব পরিমার্জিত হাই লেভেল ল্যাংগোয়েজ প্রোগ্রামিং ভাষা।

Digital—(১) সংখ্যার সাহায্যে কাজ করে এমন যন্ত্র বা পদ্ধতি।

(২) বাইনারি সংখ্যার সাহায্যে কাজ করে এমন যন্ত্র বা পদ্ধতি।

Direct Access—অন্য কোনো তথ্যের সাহায্য না নিয়ে, স্মৃতি ভাণ্ডার থেকে সরাসরি কোনো বিশেষ তথ্য খুঁজে বার করার ক্ষমতা।

Direct command—এক বা একাধিক বিশেষ নির্দেশের সমষ্টি, যে নির্দেশ কম্পিউটার তৎক্ষণাৎ পালন করে।

Disc, magnetic—কম্পিউটার তথ্য রেকর্ডিং করে রাখার ঘূর্ণন্ত চৌম্বক ডিস্ক।

Disc storage—চৌম্বক ডিস্কের মাধ্যমে কম্পিউটার স্মৃতি সংরক্ষণ ব্যবস্থা।

Diskett—floppy disc এর অপর নাম।

Display—প্রোগ্রাম, নির্দেশ এবং তথ্যকে চাক্ষুস ভাবে ফুটিয়ে তোলার যন্ত্র, যেমন L. E. D ডিসপ্লে, V. D. U, LCD ইত্যাদি।

Document—প্রোগ্রাম, তথ্য এবং নির্দেশ ধরে রাখার মাধ্যম, যেমন পাঞ্চ কার্ড, টেপ ইত্যাদি।

DOS—Disc-based Operating System কথাটির সংক্ষিপ্ত রূপ।

Down time—একটি সচল কম্পিউটার খারাপ হলে, তাকে ফের সারিয়ে সচল করতে যে সময় লাগে।

Drive—রিড্ / রাইট্ হেডের সামনে দিয়ে চৌম্বক ফিতেকে টেনে নিয়ে যাবার যান্ত্রিক ব্যবস্থা।

Dump—স্মৃতি ভাণ্ডারের তথ্যকে পর পর সঠিক ক্রমে সাজিয়ে লেখা।

Duplex—একই পরিবাহী পথ ধরে তথ্য যাওয়া এবং আসার ব্যবস্থা।

E

EBCDIC—Extended Binary Coded Decimal Interchange Code কথাটির সংক্ষিপ্ত রূপ। ASCII কোডের মতই আরএকটি জনপ্রিয় কোড, যার সাহায্যে কম্পিউটার ক্যারেকটার এবং কম্যাণ্ডকে 8 Bit বাইনারি সংখ্যার দ্বারা চিহ্নিত করা হয়।

Edit—প্রোগ্রামের বিষয় বস্তু পরিবর্তন করা।

EEPROM—বিশেষ ধরনের PROM যার স্মৃতিকে ইচ্ছা করলে বিদ্যুৎ স্পন্দন পাঠিয়ে মুছে ফেলে নূতন তথ্য লেখা যায়। পুরো কথাটি হলো Electrically Erasable and Programable Read Only Memory.

Emulator—একটি বিশেষ যন্ত্র, যার সাহায্যে কোনো কম্পিউটার, অন্য কম্পিউটারের জন্য লেখা মেশিন ল্যাংগোয়েজ প্রোগ্রাম থেকে নির্দেশ গ্রহণ করতে পারে।

End of file—স্মৃতি ভাণ্ডারে সংরক্ষিত তথ্যের শেষে দেওয়া সংকেত, যা থেকে কম্পিউটার বুঝতে পারে তথ্যটি ওই জায়গায় শেষ হয়েছে।

Enter—কম্পিউটারে একটি সম্পূর্ণ নির্দেশ বা তথ্য পাঠানো।

Erase head—চৌম্বক পদার্থে রেকর্ড করা তথ্য মুছে দেওয়ার জন্য বিশেষ ভাবে নির্মিত হেড।

Error condition—কম্পিউটারের স্বাভাবিক গণনা কার্য থেকে বিচ্যুতি।

Error message—স্বাভাবিক কাজকর্মে বিচ্যুতি ঘটলে কম্পিউটার কর্তৃক ডিসপ্লেতে ফুটিয়ে তোলা সংকেত।

Execute—কম্পিউটার কর্তৃক প্রোগ্রামে দেওয়া নির্দেশ পালন।

EXOR—Exclusive OR কথাটির সংক্ষিপ্ত রূপ। ডিজিটাল গেট বিশেষ।

Expression—কন্সট্যাণ্ট, ভেরিয়েবল এবং নির্দেশের সমষ্টি।

F

False—কোন উত্তর বা অবস্থা, যেটি কম্পিউটার কর্তৃক মিথ্যে বলে বিবেচিত। এটিকে বাইনারি 0 দ্বারা চিহ্নিত করা হয়।

Facsimile—অবিকল একই রকম নকল (true copy)।

Facsmile system—যে যন্ত্রের সাহায্যে ছবি, লেখা, ম্যাপ ইত্যাদির হুবহু নকল বৈদ্যুতিক সংকেতের সাহায্যে বহু মাইল দূরে অবস্থিত আর একটি স্থানে পাঠানো হয়।

Fetch—স্মৃতি ভাণ্ডার থেকে কোনো তথ্য আহরণ করে তাকে CPU তে বিশ্লেষণ করা।

Fibre Optic Cablie—আলোক রশ্মির সাহায্যে তথ্য আদান প্রদানের কাজে ব্যবহৃত আলোকরশ্মি পরিবহণকারী স্বচ্ছ তার। বৈদ্যুতিক তরঙ্গ যেমন ইলেকট্রিক তারের মাধ্যমে সরল বা বক্র পথে এক জায়গা থেকে অন্য জায়গায় পাঠানো যায়, ফাইবার অপটিক তারকেও বাঁকিয়ে চুরিয়ে যেমন ভাবেই রাখা হোক না কেন, তারের স্বচ্ছ মাধ্যমে আলোকের পূর্ণ আভ্যন্তরীণ প্রতিসরণের (Total internal refraction) জন্য আলোক রশ্মিটিও ওই পথ ধরে

ট্রান্সমিটার থেকে রিসিভারে তথ্য বহন করে নিয়ে যায়।

File—পরস্পর সম্বন্ধিত তথ্যের সমষ্টি।

Firmware—স্থায়ী ROM স্মৃতি ভাণ্ডারে ভরে দেওয়া প্রোগ্রাম (Software)।

Flip Flop—বিশেষ ইলেক্ট্রনিক সার্কিট্, যেটিকে ইচ্ছেমতো বাইনারি 1 কিম্বা বাইনারি 0 অবস্থায় রেখে দেওয়া যায়। এই সার্কিটকে RAM স্মৃতি ভাণ্ডারের স্মৃতি কোষ হিসেবে ব্যবহার করা হয়।

Floppy disc—চৌম্বক পদার্থের প্রলেপ লাগানো খুব পাতলা ডিস্ক, 45 R. P M রেকর্ডের মতো প্লাস্টিকের চৌকো খাপে ভরাথাকে। এই ডিস্কে কম্পিউটারের তথ্য ও প্রোগ্রাম রেকর্ড করা হয়।

Flow chart—কোন বিশেষ সমস্যা কি ভাবে যুক্তি গ্রাহ্য পথ ধরে সমাধান করতে হবে তারই সচিত্র বিবরণ।

Function—একটি প্রক্রিয়া, যাতে কম্পিউটার কেবল মাত্র এক বা একাধিক গাণিতিক মান গ্রহণ করে তার সাহায্যে একটি গাণিতিক ক্রিয়া সম্পন্ন করে এবং উত্তর বলে দেয়।

G

Garbage—অর্থহীন তথ্য।

Gate—বাইনারি অঙ্ক কষার কাজে ব্যবহৃত বিশেষ সুইচিং সার্কিট।

Graphics—কম্পিউটারের সাহায্যে ছবি আঁকার ব্যবস্থা।

H

Hard copy—প্রিন্টারের সাহায্যে কাগজে ছাপানো বিষয় বস্তু।

Hardware—কম্পিউটার এবং তার সঙ্গে লাগানো সমস্ত রকম যান্ত্রিক সংযুক্তি।

Head—তড়িৎ চুম্বকীয় যন্ত্রাংশ, যার সাহায্যে চৌম্বক পদার্থের ফিতে এবং ডিস্কে কম্পিউটারের তথ্য রেকর্ড হয়, আর পূর্বে রেকর্ডিং করা তথ্য পড়া হয়।

Hexadecimal—ষোলটি সংখ্যা নিয়ে গঠিত সংখ্যা পদ্ধতি, যার র‍্যাডিক্স হলো ষোল। অ্যাসেম্বলি ল্যাংগোয়েজে কম্পিউটার প্রোগ্রাম করতে এই সংখ্যা পদ্ধতি ব্যবহার করা হয়।

Hierarchy—কোন বিষয় সমষ্টির ক্রম বিন্যাস।

Holographic lens—ডিফ্র্যাকশান গ্রেটিং (Diffraction Grating) দ্বারা নির্মিত বিশেষ লেন্স। এর সাহায্যে লেসার রশ্মিকে বিভিন্ন দিকে এবং বিভিন্ন দূরত্বে ঘোরানো ও ফোকাস করা যায়।

Hopper—পানচ্ কার্ড রিডার যন্ত্রে কার্ড রাখার জায়গা।

Hybrid computer—বিশেষ ধরনের কম্পিউটার, যেটি অ্যানালগ এবং ডিজিটাল, উভয় ধরনের সংকেতই বুঝতে পারে।

I

IBM—International Business Machine। পৃথিবীর বৃহত্তম কম্পিউটার নির্মাতা।

Idle time—কম্পিউটারে কোন গণনা কার্য না চলাকালিন সময়।

Impact printer—যে সমস্ত কম্পিউটার প্রিন্টারে কাগজের ওপর কালির ফিতেকে ধাতব টাইপ অথবা পিনের সাহায্যে চাপ দিয়ে লেখা ফুটিয়ে তোলা হয়।

Initialize—কোনো প্রোগ্রাম শুরুর আগে সমস্ত ভেরিয়েবলকে একটি প্রাথমিক মান দেওয়া।

Ink jet printer—কাগজের ওপর কালির ফোয়ারার সাহায্যে ফুটকি এঁকে এঁকে লেখা ছাপানোর যন্ত্র।

Input—প্রোগ্রাম, নির্দেশ বা তথ্যকে ইনপুট ডিভাইস কিম্বা বাড়তি স্মৃতি ভাণ্ডার থেকে আভ্যন্তরীণ স্মৃতি ভাণ্ডারে পাঠানোর প্রক্রিয়া।

Instruction—কম্পিউটারকে কোন বিশেষ কাজ করতে বলার সাংকেতিক নির্দেশ (Keyword)।

Integer—পূর্ণ সংখ্যা, যেমন 56, – 4, 1 ইত্যাদি।

Integrated circuit—বড় ইলেক্ট্রনিক সার্কিটকে সিলিকন পাতের ওপর অতি ক্ষুদ্র আকারে তৈরী করার পদ্ধতি।

Intelligent terminal—কোনো একটি বিশেষ কাজের জন্য নির্মিত কম্পিউটার।

Interface—বিশেষ ইলেক্ট্রনিক সার্কিট, যেটি মূল কম্পিউটারের সঙ্গে ইনপুট, আউটপুট, বাড়তি স্মৃতি ভাণ্ডার এবং অন্য কম্পিউটারকে সংযুক্ত করে।

Interpreter—(১) কম্পিউটারের একটি প্রোগ্রাম, যার সাহায্যে C. P. U, হাই লেভেলে লিখিত প্রোগ্রামকে মেশিন ল্যাংগোয়েজে অনুবাদ করতে পারে।

(২) পাঞ্চ কার্ডের ফুটো সংকেত পড়ে তাতে কি প্রোগ্রাম লেখা আছে সেটি ছাপিয়ে দেওয়ার যন্ত্র।

Interrupt—কম্পিউটারের স্বাভাবিক গণনা কার্যকে প্রোগ্রামার কর্তৃক ইচ্ছাকৃত ভাবে থামিয়ে দেওয়া।

I/O—Input/Output কথাটির সংক্ষিপ্ত রূপ।

J

Jargon—সাধারণ মানুষের কাছে দুর্বোধ্য কম্পিউটার পরিভাষা।

Joystick—ভিডিও গেম খেলাতে খেলোয়াড়ের অবস্থান নিয়ন্ত্রণের জন্য ব্যবহৃত লিভার লাগানো ইলেক্ট্রনিক যন্ত্র।

Jump—কম্পিউটার কর্তৃক প্রোগ্রামে লিখিত নির্দেশগুলি সঠিক ক্রমে পালন না করে অন্য নির্দেশ পালন করা।

K

K—kilo কথাটির সংক্ষিপ্ত রূপ। 1 k বলতে বোঝায় 1000।

Key Board—টাইপ রাইটারের মতো দেখতে বিভিন্ন ক্যারেকটার এবং কম্যাণ্ড লেখা যন্ত্র। এর বোতাম টিপে কম্পিউটারে প্রোগ্রাম করা হয়।

Key punch—কার্ড কিম্বা টেপে ফুটো পানচ্ করার যন্ত্র।

Keyword—Instruction দেখুন।

L

Language—কম্পিউটার প্রোগ্রাম করার ভাষা।

Large Scale Integration—Integrated circuit প্রযুক্তির থেকেও ইলেক্ট্রনিক সার্কিটকে আরো ক্ষুদ্রায়িত করার পদ্ধতি।

Laser—Light Amplification by Stimulated Emission of Radiation কথাটির সংক্ষিপ্তরূপ। অত্যন্ত শক্তিশালী একবর্ণী (Monochromatic) সুসংবদ্ধ (Coherent) এবং সমবর্তিত (Polarized) আলোক রশ্মি তৈরী করার পদ্ধতি।

Library Function—পূর্ব ব্যাখ্যাত প্রোগ্রাম এবং রুটিনের সমষ্টি।

Light pen—পেনের মতো দেখতে একটি ইলেক্ট্রনিক যন্ত্র, যেটিকে মনিটর স্ক্রিনের ওপর ধরলে মনিটর স্ক্রিনের ওই অবস্থানটির তথ্য কম্পিউটার জেনে যায়।

Line printer—যে প্রিন্টারে একবারে একটি পুরো লাইন ছাপা হয়।

Lode—বাড়তি স্মৃতি ভাণ্ডার থেকে কম্পিউটারে প্রোগ্রাম বা তথ্য ভরার কাজ।

Log—Logarithm শব্দের সংক্ষিপ্ত রূপ।

Logic—কোন ছুটি ঘটনার মধ্যে যুক্তি গ্রাহ্য সম্পর্ক স্থাপন।

Loop—প্রোগ্রামের একটি অংশ, যাকে দুই বা ততোধিক বার পালন করা হয়।

M

Machine Code—কম্পিউটারের C. P. U যে ভাষায় নির্দেশ বুঝতে পারে।

Magnetic Ink Character Recognition (MICR)—চৌম্বক কালি দিয়ে বিশেষ হরফে ছাপা প্রোগ্রাম থেকে তথ্য পড়ার ব্যবস্থা।

Mainframe—16 কিম্বা 32 বিট্ তথ্য এক সঙ্গে আদান প্রদান করতে পারে এমন ধরনের বড় কম্পিউটার।

Masterfile—বাড়তি স্মৃতিভাণ্ডারে রক্ষিত সেই ধরনের তথ্য, যেটি খুব বেশী অদলবদল করা হয়না, বরং মাঝে মধ্যে update করা হয়। (update দেখুন)।

Matrix—Array দেখুন।

Memory—স্মৃতি ভাণ্ডার।

Memory Map—কম্পিউটারের আভ্যন্তরীণ ROM এবং RAM স্মৃতি ভাণ্ডারে বিভিন্ন কাজের জন্য পূর্বনির্ধারিত মেমরি সেল ব্যবস্থা।

Merge—দুই বা ততোধিক ফাইলকে একত্রিত করা।

Message—একত্রিত করে পাঠান হয় এমন শব্দ সমষ্টি।

Micro Computer—মাইক্রো প্রসেসর দিয়ে নির্মিত 2, 4 কিম্বা 8 বিট্ কম্পিউটার।

Microprocessor—VLSI প্রযুক্তির সাহায্যে নির্মিত বিশাল জটিল সার্কিট সমূহের অতি ক্ষুদ্রায়িত রূপ। বর্তমানে এগুলিই হলো অতিক্ষুদ্রায়নের চূড়ান্ত নিদর্শণ।

Micro second—এক সেকেণ্ডের এক লক্ষ ভাগের এক ভাগ (10^{-6} সেকেণ্ড)।

Milli second—এক সেকেণ্ডের এক হাজার ভাগের এক ভাগ (10^{-3} সেকেণ্ড)।

Mini Computer—মাইক্রো এবং মেইনফ্রেম কম্পিউটারের অন্তবর্তী শ্রেণীর কম্পিউটার। সাধারণত একসঙ্গে 16 বিট্ তথ্য আদান প্রদানে সক্ষম কম্পিউটারকেই মিনি কম্পিউটার বলে অভিহিত করা হয়।

Mnemonic—মেশিন ল্যাংগোয়েজে প্রোগ্রাম করার সময় ব্যবহৃত সাংকেতিক শব্দ, যার সাহায্যে প্রোগ্রামার সহজে বুঝতে পারেন বা মনে রাখতে পারেন কম্পিউটারকে কি নির্দেশ দেওয়া হচ্ছে। উচ্চারণটি হলো নেমোনিক্।

Modem—Modulators demodulators কথাটির সংক্ষিপ্ত রূপ। এই যন্ত্রটি কম্পিউটারের ডিজিটাল স্পন্দন থেকে অ্যানালগ তরঙ্গ তৈরী করতে পারে, আবার ইচ্ছা করলে অ্যানালগ তরঙ্গ থেকে ডিজিটাল স্পন্দনও তৈরী করতে পারে। কম্পিউটারের সঙ্গে মডেম লাগিয়ে কম্পিউটারের তথ্য সাধারণ টেলিফোন লাইনের মাধ্যমে যেখানে খুশি পাঠানো যায়।

Monitor—V. D. U. দেখুন

Mouse—ইলেক্ট্রনিক যন্ত্র বিশেষ। এর সাহায্যে V. D. U. স্ক্রিনে cursor কে যেখানে খুশি সরানো যায়।

Multiplexing—একাধিক তথ্যের উৎস থেকে একটি তারের মাধ্যমে পর্যায়ক্রমে তথ্য গ্রহণ করা কিম্বা একাধিক আউটপুট ডিভাইসে তথ্য জোগান দিয়ে যাওয়া।

Multiprogramming—একইসঙ্গে একাধিক প্রোগ্রাম করার পদ্ধতি।

N

NAND—NOT AND কথাটির সংক্ষিপ্ত রূপ। এটি একটি ডিজিটাল গেট।

Nano second—এক সেকেণ্ডের এক কোটি ভাগের এক ভাগ (10^{-9} সেকেণ্ড)।

Natural language—মানুষের বোধ্য ভাষা, যেমন বাংলা, ইংরেজী, হিন্দি ইত্যাদি।

Nesting—একই প্রোগ্রামের মধ্যে একাধিক লূপ প্রোগ্রাম এমন ভাবে ব্যবহার করা, যাতে একটি লূপের মধ্যে অন্য একটি বা একাধিক লূপ স্থাপিত হয়।

Net work—একাধিক পরস্পর সংযুক্ত কম্পিউটার মিলিয়ে গঠিত একটি বড় কম্পিউটার ব্যবস্থা।

Nibble—4 বিট্ (Bit) তথ্য অথবা 1/2 বাইট তথ্য।

Non Volatile—স্থায়ী স্মৃতি ব্যবস্থা। এর বৈদ্যুতিক জোগান বন্ধ করলেও লিখিত স্মৃতি মুছে যায় না।

NOR—NOT OR কথাটির সংক্ষিপ্ত রূপ। এটি একটি ডিজিটাল গেট্।

NOT—ডিজিটাল গেট বিশেষ (Inverter)।

Notation—কোন বস্তুর প্রতীক হিসেবে ব্যবহৃত বিশেষ চিহ্ন বা ক্যারেকটার।

Numeric Variable—যে সব ভেরিয়েবলের গাণিতিক মান আছে।

O

Optical Character Recognition (OCR) —বিশেষ ইনপুট ব্যবস্থা যেটি ছাপানো হরফ কিম্বা হাতের লেখা পড়তে পারে।

Octal—8 র‍্যাডিক্স যুক্ত সংখ্যা পদ্ধতি। এতে মোট আটটি সংখ্যা আছে 0, 1, 2, 3, 4, 5, 6, 7. কম্পিউটার প্রোগ্রাম লেখার সময় অনেক ক্ষেত্রে ব্যবহার করা হয়।

Off line—কম্পিউটারের সঙ্গে জড়িত অথচ C. P. U. দ্বারা নিয়ন্ত্রিত নয় এমন যন্ত্র সংক্রান্ত কোনো ব্যাপার।

On line—C. P. U. দ্বারা নিয়ন্ত্রিত যন্ত্র সংক্রান্ত কোনো ব্যাপার।

Operand—কম্পিউটারকে যে নির্দেশ দেওয়া হচ্ছে, সেই নির্দেশটির Memory address. সাধারণত assembly language ভাষায় ব্যবহার করা হয়।

Operation code—assembly language এ লিখিত নির্দেশ, যেটি কম্পিউটার পালন করবে। একে সংক্ষেপে Opcode ও বলা হয়।

Operating system—সামগ্রিক ভাবে কম্পিউটারের কাজকর্ম নিয়ন্ত্রণকারী Software বা প্রোগ্রাম।

Operator—বিশেষ নির্দেশ, যেটি গাণিতিক কিম্বা লজিকাল অপারেশন করায়।

Order—নির্দেশ কিম্বা ঘটনার পর্যায়ক্রম।

OR—বিশেষ লজিক গেট।

Output—গণনা কার্যের পর কম্পিউটারের উত্তর।

Output device—যে যন্ত্রের মাধ্যমে কম্পিউটার উত্তর প্রকাশ করে।

Overflow—গণনার পর উত্তরটি কম্পিউটরের উত্তর প্রকাশ ক্ষমতার বেশী হয়ে যাওয়া।

P

Pack—প্রোগ্রাম করার আগে সঠিক ক্রমে সাজানো পানচ্ কার্ডের স্তূপ।

Pad (১) —সার্কিট বোর্ডে তার ঝালাই করার জন্য তামার টার্মিনাল। (২) কী বোর্ড।

Paper tape—কাগজের তৈরী লম্বা ফিতে, যাতে ফুটো করে কম্পিউটার প্রোগ্রামকে লিপি বদ্ধ করা হয়।

Paper tape reader—পানচ্ টেপে লিপিবদ্ধ তথ্য পড়ার যন্ত্র।

Perallel interface—যে interface ব্যবস্থায় কম্পিউটার এবং পেরিফেরালসের (Peripherals) মধ্যে এক সঙ্গে 8 Bit তথ্য আদান-প্রদান করা হয়।

PASCAL—হাইলেভেল প্রোগ্রামিং ভাষা। বিখ্যাত ফরাসী গণিতজ্ঞ বেইজ পাসকাল এর নামানুসারে এর নামকরণ। 1970 সালে সুইজারল্যাণ্ডে নিকোলাস রিথ এই ভাষা তৈরী করেন। FORTRAN-এর থেকে সহজ আর BASIC-এর থেকে শক্তিশালী এই সংক্ষিপ্ত প্রোগ্রামিং ভাষা মিনি কম্পিউটারে বেশী ব্যবহার করা হয়।

Pass word—গোপন তথ্য ফাইলের নিরাপত্তার জন্য, বাড়তি স্মৃতি ভাণ্ডারে সংরক্ষিত ফাইল থেকে তথ্য পড়ার আগে যে সংকেত বার্তা কম্পিউটারকে জানতে হয়।

Path—কোন প্রোগ্রাম অনুসারে কাজ করার সময় কম্পিউটার যে যুক্তি গ্রাহ্য পথ ধরে এগোয়।

Peripherals—মূল কম্পিউটারের সঙ্গে বাহ্যিক ভাবে যে সমস্ত বাড়তি যান্ত্রিক সংযুক্তি (attachment) লাগানো হয়, যেমন প্রিন্টার, মনিটর, মডেম ইত্যাদি।

Petal printer—Daisy Wheel প্রিন্টারের অপর নাম।

Pico second—এক সেকেণ্ডের এক হাজার কোটি ভাগের এক ভাগ (10^{-12} সেকেণ্ড)।

Pixel—মনিটর স্ক্রিনে ফুটে ওঠা ক্ষুদ্রতম আলোক বিন্দু।

PL/M—IBM কর্তৃক সৃষ্ট হাই লেভেল প্রোগ্রামিং ভাষা। FORTRAN এর থেকে শক্তিশালী। একমাত্র বড় মেইন ফ্রেম কম্পিউটারেই ব্যবহার করা হয়।

PL/1—PL/1 এর সংক্ষিপ্ত রূপ। INTEL কোম্পানী এই ভাষাটি সৃষ্টি করেছেন।

Plotter—কম্পিউটার গ্রাফিক্স ছাপার জন্য ব্যবহৃত বিশেষ ধরনের প্রিণ্টার।

Port—কম্পিউটারের সঙ্গে বিভিন্ন পেরিফেরালস্ লাগানোর জন্য বৈদ্যুতিক সকেট।

Printer—কম্পিউটারের উত্তর ছাপাবার যন্ত্র।

Processor—C. P. U. দেখুন।

Program—কত গুলি নির্দেশের সমষ্টি, যেটি কম্পিউটারের C. P. U. কে কোন একটি বিশেষ কাজ বা সমস্যা কি ভাবে সমাধান করতে হবে, বলে দেয়।

Progammer—যিনি কম্পিউটারের প্রোগ্রামটি তৈরী করেন।

Programming Language—যে ভাষায় কম্পিউটারকে পাঞ্চ কার্ড, টেপ, কী বোর্ড ইত্যাদির মাধ্যমে প্রোগ্রাম করা হয়।

PROM—Programable Read Only Memory কথাটির সংক্ষিপ্ত রূপ। এটি একটি ইলেক্ট্রনিক স্থায়ী স্মৃতি ব্যবস্থা।

Punch card—card দেখুন।

Punch card reader—কার্ড থেকে তথ্য পড়ার যন্ত্র।

Prototype—বাণিজ্যিক উৎপাদনের আগে গবেষণাগারে প্রস্তুত পরীক্ষা মূলক যন্ত্র।

R

Radix—কোনো সংখ্যা পদ্ধতির Radix নির্দেশ করে সেই সংখ্যা পদ্ধতিতে কটি মূল সংখ্যা আছে। যেমন দশমিক পদ্ধতির Radix 10, কারণ দশমিক পদ্ধতিতে 0,1,2,3,4,5,6,7,8,9 এই দশটি মূল সংখ্যা আছে।

RAM—Random Access Memory কথাটির সংক্ষিপ্ত রূপ। এটি হলো ইলেক্ট্রনিক অস্থায়ী স্মৃতি ব্যবস্থা।

Random Access—Direct Access দেখুন।

Range—কোনো ভেরিয়েবল বা ফাংশনের সকল সম্ভাব্য মান।

Raster—মনিটর স্ক্রিনে ইলেক্ট্রন রশ্মি Scan করে যাবার দরুন যে দাগ হয়।

Read—স্মৃতি ভাণ্ডার থেকে তথ্য পড়া।

Read/write head—চৌম্বক মাধ্যমে তথ্য লেখা বা পড়ার জন্য নির্মিত তড়িৎ চুম্বকীয় যন্ত্রাংশ।

Real time—অত্যন্ত দ্রুত গণনা করতে পারে এমন on line ব্যবস্থা।

এই ধরনের যন্ত্রের সাহায্যে কোনো প্রক্রিয়ার (রাসায়নিক পদার্থ উৎপাদন, কলকারখানার যন্ত্রপাতি নিয়ন্ত্রণ ইত্যাদি) ওপর নজর রাখা হয় যাতে উৎপাদনের পরিমাণ সর্বাধিক হয় কিম্বা প্রক্রিয়াটি সঠিক হয়।

এই ব্যবস্থা কোনো বাহ্যিক প্রক্রিয়া থেকে তথ্য গ্রহণও বিশ্লেষণ করে তার ফলাফল অনুযায়ী প্রক্রিয়াটিকে নিয়ন্ত্রণ করে। পুরো ব্যাপারটি এতো দ্রুত ঘটে যে প্রক্রিয়া থেকে তথ্য আহরণ এবং সেই অনুসারে প্রক্রিয়া নিয়ন্ত্রণের নির্দেশ প্রায় একই সময়ে ঘটে।

Register—C. P. U. এর অন্তর্গত ছোট RAM স্মৃতি ভাণ্ডার।

Relay—তড়িৎ চুম্বকের সাহায্যে চালিত বৈদ্যুতিক স্যুইচ্।

Rerun—আগের প্রোগ্রাম অনুসারে কম্পিউটারকে পুনরায় চালনা করা।

Resistor—বিদ্যুৎ প্রবাহকে বাধা প্রদানকারী বস্তু (রোধক)।

Resolution—কম্পিউটার গ্রাফিক্সে সব থেকে সূক্ষ্ম বিষয় বস্তু আঁকার ক্ষমতা।

Retrieve—স্মৃতি ভাণ্ডার থেকে কোন একটি বিশেষ তথ্য খুঁজে বার করা।

Rewind—চৌম্বক ফিতেকে পুনরায় শুরু হওয়ার জায়গায় নিয়ে আসা।

Robot—কম্পিউটার নিয়ন্ত্রিত চলাফেরা এবং যান্ত্রিক হাতের সাহায্যে বিভিন্ন জিনিস এক জায়গা থেকে অন্য জায়গায় সরিয়ে রাখার ক্ষমতা সম্পন্ন যন্ত্র।

Robotics—Robot নির্মাণ সংক্রান্ত বিদ্যা।

ROM—Read Only Memory কথার সংক্ষিপ্ত রূপ। এটি একটি স্থায়ী ইলেক্ট্রনিক স্মৃতি ভাণ্ডার।

S

Save—কম্পিউটারের আভ্যন্তরীণ অস্থায়ী স্মৃতি ভাণ্ডারে রক্ষিত তথ্যকে বাড়তি স্মৃতি ভাণ্ডারে স্থায়ী ভাবে সংরক্ষণ।

Scroll—বিশেষ ব্যবস্থা, যার সাহায্যে প্রোগ্রামার ইচ্ছা করলে স্ক্রিনে ফুটে থাকা বিষয় বস্তুকে নিচ থেকে ওপরে বা পাশাপাশি সরিয়ে নিয়ে যেতে পারেন। এর ফলে পুরোনো তথ্য ক্রমশ স্ক্রিন থেকে বেরিয়ে যায় এবং নতুন তথ্য স্ক্রিনের ভিতর প্রবেশ করে।

Seek—ফ্লপি কিম্বা হার্ড ডিস্কে রেকর্ড করা তথ্য খুঁজে বার করার জন্য Read/Write হেডের নড়াচড়া।

Semiconductor—বৈদ্যুতিক ভাবে পরিবাহী নয় আবার অপরিবাহীও নয় এমন কঠিন বা তরল পদার্থ। এদের অর্ধপরিবাহী বলা হয়। অর্ধপরিবাহীর বৈশিষ্ট্য হলো, এদের ইলেক্ট্রন পরিবাহিতা তাপমাত্রা বৃদ্ধির সঙ্গে সঙ্গে বৃদ্ধি পায়। তাছাড়া পদার্থটির বিশুদ্ধতার ওপরও পরিবাহিতা বহুলাংশে নির্ভর করে।

সমস্ত সক্রিয় ইলেক্ট্রনিক যন্ত্রাংশ এই অর্ধপরিবাহী পদার্থ দিয়ে তৈরী হয়। ইলেক্ট্রনিক্স প্রযুক্তিতে সব থেকে বেশী ব্যবহৃত কয়েকটি অর্ধপরিবাহী হলো—জার্মেনিয়াম (Germanium), সিলিকন (Silicon), সেলিনিয়াম (Selenium), কিউপ্রাস অক্সাইড (Cuprous Oxide), গ্যালিয়াম আর্সেনাইড (Gallium Arsenide), লেড টেলুরাইড (Lead Telluride) প্রভৃতি।

Sequence—একাধিক তথ্যকে সঠিক ক্রমে সাজানো।

Serial—একটি Data Bus এর মাধ্যমে ডিজিটাল সংকেত আদান প্রদান করা।

Serial interface—যে interface ব্যবস্থায় কম্পিউটার ও পেরিফেরালস্ এর মধ্যে একটি তারের সাহায্যে বাইনারি তথ্য আদান প্রদান হয়।

Set—একত্রিত করা একই ধরনের বস্তু বা তথ্য।

Sign—ধনাত্মক বা ঋণাত্মক সংখ্যার পরিচয়জ্ঞাপক চিহ্ন (+ এবং –)।

Significant digit—কোন সংখ্যার সঠিক মান জানার জন্য যে অঙ্কটি (digit) জানা প্রয়োজন। একাধিক অঙ্ক (digit) বিশিষ্ট কোন সংখ্যার (number) সব থেকে বাঁ দিকের অঙ্কটিকে Most significant digit এবং সব থেকে ডান দিকের অঙ্কটিকে Least significant digit বলা হয়।

Silicon—অর্ধপরিবাহী মৌলিক পদার্থ। সিলিকা অর্থাৎ বালি থেকে বিশেষ বিশুদ্ধকরণ ও কেলাসন পদ্ধতিতে এটি উৎপাদন করা হয়। সিলিকন দিয়ে সেমিকণ্ডাকটর ডায়োড, ট্রানজিষ্টর, থাইরিস্টর প্রভৃতি সক্রিয় ইলেক্ট্রনিক যন্ত্রাংশ তৈরী হয়। তাছাড়া এরই অতি পাতলা ছোট টুকরোর ওপর IC, MSI, LSI এবং VLSI তৈরী করা হয়।

Simulation—কোন একটি বিশেষ যন্ত্র বা পদ্ধতি হাতে কলমে তৈরী না করে, সেই বিষয়ে কম্পিউটারে যাবতীয় তথ্য ভরে দিয়ে কম্পিউটারের সাহায্যে কৃত্রিম ভাবে সেই যন্ত্রটি বা পদ্ধতিটির আচার আচরণ সৃষ্টি করা।

Slide rule—লগারিদ্‌ম গণিতের সাহায্য নিয়ে প্রস্তুত গণনা সহায়ক যন্ত্র।

Software—কম্পিউটারের যে কোনো প্রোগ্রাম (ROM এবং অন্যান্য স্থায়ী স্মৃতিতে লিপিবদ্ধ প্রোগ্রামকেও Software বলা হয়)।

Solid state—যে সমস্ত ইলেক্ট্রনিক যন্ত্রাংশ অর্ধপরিবাহী পদার্থ দিয়ে তৈরী করা হয়।

Space—প্রোগ্রামে লিখিত কোনো দুটি ক্যারেক্টার বা শব্দকে আলাদা করে দেখানোর জন্য যে ফাঁক রাখা হয় ;

Special Character—হরফ বা সংখ্যা ছাড়া কম্পিউটারের বোধ্য অন্যান্য ক্যারেকটার। যেমন *, /, ?, !, & প্রভৃতি।

Speech Synthesizer—বিশেষ ডিজিটাল এবং অ্যানালগ মিশ্র সার্কিট। এটি কম্পিউটারের আউটপুট সিগন্যাল অনুসারে বিভিন্ন দৈর্ঘ্যের এবং উচ্চারণের শব্দ করতে পারে। এই শব্দ গুলিকে ফোনেম (Phonem) বলা হয়। কম্পিউটার প্রোগ্রামার ফোনেম চার্ট দেখে কম্পিউটারে এমন প্রোগ্রাম করতে পারেন যাতে কম্পিউটার, স্পিচ্‌ সিন্থেসাইজারকে নিয়ন্ত্রণ করে পরপর কতগুলি ফোনেম সৃষ্টি করে এবং এই ফোনেমগুলি এর সঙ্গে বাজলে মানুষের গলার আওয়াজে ইংরেজী শব্দের সৃষ্টি হয়। এই ভাবে কম্পিউটারের মাধ্যমে মানুষের গলার আওয়াজ তৈরী করা যায়। বর্তমানে বেশ কিছু ইলেক্ট্রনিক ঘড়ি এবং খেলনা রোবটে কম্পিউটার নিয়ন্ত্রিত স্পিচ্‌ সিন্থেসাইজার ব্যবহার করা হচ্ছে।

Statement—প্রোগ্রামে লিখিত কম্পিউটারের যে কোনো নির্দেশ।

Step—প্রোগ্রামে লিখিত কোন একটি নির্দেশ।

Storage—কম্পিউটার তথ্য জমিয়ে রাখার যে কোনো স্মৃতি ভাণ্ডার।

String—কোটেশন চিহ্নের মধ্যে লিখিত একাধিক হরফ বা ক্যারেকটারের সমষ্টি।

String Variable—যে ভেরিয়েবলের কোনো গাণিতিক মান নেই।

Super computer—জটিল বৈজ্ঞানিক সমস্যা সমাধানের জন্য অত্যন্ত দ্রুতগতি সম্পন্ন বিশাল কম্পিউটার।

Supervisor—ROM স্মৃতিতে রক্ষিত বিশেষ Software. এটির সাহায্যে C. P. U. প্রোগ্রামারের নির্দেশ ছাড়াই বাড়তি স্মৃতি থেকে তথ্য পড়তে পারে, প্রোগ্রামের বিভিন্ন ভুল ত্রুটি খুঁজে বার করতে পারে এবং অন্যান্য পেরিফেরালসদের নিয়ন্ত্রণ করতে পারে।

Syntax—হাই লেভেল ল্যাংগোয়েজে প্রোগ্রাম লেখার সময় বিভিন্ন Keyword, Constant, Variable এবং অন্যান্য expression কে সাজানোর সঠিক ক্রম।

Syntax error—প্রোগ্রাম লেখার সময় Syntax গত ত্রুটি।

System analysis—কম্পিউটারের সাহায্যে কোনো বাণিজ্যিক উৎপাদন বা পদ্ধতির প্রতিটি ধাপ বিশ্লেষণ করে কিভাবে উৎপাদন বাড়ানো যায় তার ফলাফল নিরূপণ করা।

T

Table—তথ্যের তালিকা (Array)।

Tag—কোনো তথ্য, যাকে নির্দেশিকা (Marker) কিম্বা লেবেল (Lable) হিসেবে ব্যবহার করা হয়।

Tape—(১) কাগজের তৈরী ফিতে, যাতে

ফুটো করে প্রোগ্রাম লেখা হয় (Punch Tape দেখুন)।

(২) চৌম্বক ফিতে, যাতে চৌম্বক ক্ষেত্রের সাহায্যে কম্পিউটার তথ্য রেকর্ড করে রাখা যায়।

Tele text—বিশেষ ইলেক্ট্রনিক ব্যবস্থা, যার সাহায্যে টেলিভিশন সেটকে টেলিফোনের সঙ্গে জুড়ে একটি বিশেষ নাম্বার ডায়াল করলে স্ক্রিনে নানা রকম প্রয়োজনীয় খবর ফুঠে উঠে।

Terminal—Console দেখুন।

Thermal printer—বিশেষ ভাবে নির্মিত, উত্তাপে রং পাল্টায় এমন কাগজে উত্তপ্ত পিনের সাহায্যে লেখা ছাপানো যন্ত্র।

Time, object/run—কোন প্রোগ্রাম অনুসারে সমস্যা সমাধান করতে কম্পিউটার যে সময় নেয়।

Timesharing—একই C. P. U. এর মাধ্যমে একই সময়ে একাধিক terminal এর সাহায্যে সম্পূর্ণ আলাদা আলাদা প্রোগ্রাম চালানোর ব্যবস্থা। এই ব্যবস্থায় একাধিক কম্পিউটার ব্যবহারকারী একটি দ্রুত গণনা সক্ষম C. P. U.-র সঙ্গে Multiplesing কায়দায় সংযুক্ত থাকেন। প্রতি কম্পিউটার ব্যবহারকারী প্রতি সেকেণ্ডে কয়েক মিলি সেকেণ্ডের জন্য মূল C. P. U. সঙ্গে সংযুক্ত হন। C. P. Uটি অত্যন্ত দ্রুত গণনা করতে পারে বলে ওই সামান্য সময়ের মধ্যেই সংযুক্ত টারমিনাল থেকে আগত তথ্য পড়ে নিয়ে বিশ্লেষণ করে উত্তর বলে দেয় আর কম্পিউটার ব্যবহারকারীর মনে হয় তিনি সারাক্ষণই মূল কম্পিউটারের সঙ্গে সংযুক্ত আছেন।

Track—ম্যাগনেটিক ডিস্কে যে বৃত্তাকার পথে তথ্য লেখা হয়।

Transistor—সিলিকন বা জার্মেনিয়ামের তৈরী তিনটি টারমিনাল যুক্ত ইলেক্ট্রনিক সুইচ্।

True—কোন উত্তর বা অবস্থা, যেটি কম্পিউটার কর্তৃক সত্য বলে বিবেচিত। এটিকে বাইনারি 1 দ্বারা চিহ্নিত করা হয়।

U

TTL—Transistor Transistor Logic কথাটির সংক্ষিপ্ত রূপ। IC নির্মাণের বিশেষ প্রযুক্তি।

Up date—বাড়তি স্মৃতি ভাণ্ডারে রক্ষিত ফাইলের (Data file) পুরোনো তথ্য মুছে সাম্প্রতিকতম তথ্য লেখা।

U. P. C.—Universal Product Code কথাটির সংক্ষিপ্ত রূপ। Bar code দেখুন।

User friendly—বিশেষ ভাবে লিখিত Software, যেটি কম্পিউটারকে বিভিন্ন পরিস্থিতিতে VDU স্ক্রিনে বিভিন্ন রকম বাক্য ফুটিয়ে তুলতে নির্দেশ দেয়। এর ফলে কম্পিউটার পরি-চালকের মনে হয় কম্পিউটারটি V.D.Uর মাধ্যমে তাঁর সঙ্গে কথোপোকথন চালাচ্ছে। যাঁরা ইতিপূর্বে কম্পিউটার ব্যবহার করেননি অথবা বাচ্চাদের ক্ষেত্রে এ ধরনের ফট্ওয়্যার কম্পিউটার। প্রোগ্রামিং শেখার ব্যাপারে আগ্রহ সৃষ্টি করে। তাছাড়া User friendly কম্পিউটার ব্যবহার করতেও অনেক বেশী আনন্দ পাওয়া যায়।

Utility Program—কম্পিউটার প্রোগ্রামিংকে আরো সহজতর করে তোলার জন্য বিশেষ Software.

V

Vacuum tube—ইলেক্ট্রনিক ভাল্বের অপর নাম।

Value—যে কোনো সংখ্যা বা String, যেটি একটি Constant বা Variable দ্বারা চিহ্নিত।

Variable—প্রোগ্রামে লিখিত কোনো হরফ বা সংখ্যা, যার মান কম্পিউটারের সাহায্যে সেই প্রোগ্রাম সমাধান করা কালীন পরিবর্তিত হয়।

V. D. U.—Video Display Unit কথাটির সংক্ষিপ্ত রূপ। এটি একটি টেলিভিশন সেটের মতো যন্ত্র, যার স্ক্রিনে কাম্পিউটার সৃষ্ট সমস্ত ক্যারেকটারের ছবি ফুটে ওঠে। আউটপুট ডিভাইস হিসেবে এর ব্যবহার সর্বাধিক।

Volatile storage—বিদ্যুৎ সরবরাহ বন্ধ হলেই যে ইলেক্ট্রনিক স্মৃতি ভাণ্ডারের সমস্ত তথ্য সম্পূর্ণ মুছে যায়, সেই স্মৃতি ভাণ্ডারকে Volatile storage বলা হয়। RAM হলো একটি volati le স্মৃতি।

W

Winchester technology—চৌম্বক পদার্থের প্রলেপ লাগানো শক্ত ধাতব চাকতিতে কম্পিউটার প্রোগ্রাম রেকর্ড করে রাখার পদ্ধতি। এই পদ্ধতিতে চাকতি বা Disc এতো জোরে ঘোরে, যে চাকতির গায়ে লেগে থাকা ঘূর্ণায়মান বাতাসের স্তরের চাপে Read/write হেডটি চাকতির ওপর ভেসে থাকে।

Disc এবং Read/write হেডের মধ্যে যান্ত্রিক সংযোগ না থাকার ফলে ঘর্ষণজনিত ক্ষয় হয় না এবং অত্যন্ত অল্প জায়গায় অনেক বেশী তথ্য রেকর্ড করা সম্ভব হয়।

Word—একটি সম্পূর্ণ তথ্য বা নির্দেশ

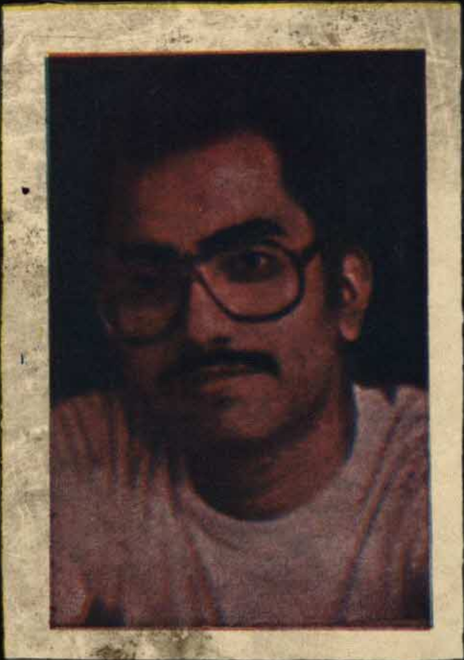
Word length—এক word তথ্যে যত সংখ্যক ক্যারেকটার বা বাইনারি Bit থাকে।

Word time—কোনো একটি স্মৃতি ভাণ্ডার থেকে অপর স্মৃতি ভাণ্ডারে এক Word তথ্য পাঠাতে যে সময় লাগে।

Working Storage—গণনাকার্যের সময় অন্তর্বর্তী উত্তরকে অস্থায়ী ভাবে সংরক্ষণের জন্য RAM স্মৃতি ভাণ্ডার। Accumulator দেখুন।

Write—C. P. U. থেকে আগত তথ্য স্মৃতি ভাণ্ডারে লিখে রাখা।

Word Processing—কী বোর্ড, মনিটর, কম্পিউটার এবং প্রিন্টারের সাহায্যে কোন রচনা বা লেখাকে ছাপানো, সংশোধন করা অথবা পুণর্বিন্যাস করা।



কম্পিউটার শিক্ষার্থীর জন্য ইংরেজি ভালো বইয়ের আজ আর অভাব নেই। কম্পিউটার সাক্ষরতার কথা মনে রেখে আমরা বাঙলা ভাষায় তেমনিই একটি উচ্চমানের সর্বাঙ্গসুন্দর সামগ্রিক আলোচনা গ্রন্থ প্রকাশনার প্রয়োজন অনুভব করি। আমাদের লক্ষ্য ছিল, বইটির ভাষা হবে সুখপাঠ্য ও সহজবোধ্য, প্রতিটি আলোচনা সচিত্র এবং শিক্ষণীয় বিষয়গুলি গ্রন্থলেখকের দ্বারা হাতে কলমে পরীক্ষিত হওয়ার পরেই হবে লিখিত। ঠিক এমনটি না হলে এ ধরনের প্রাযুক্তিক কিয়াবলম্বিত বই একান্তভাবে একাডেমিক এবং অন্য গ্রন্থনির্ভর হ'য়ে পড়ে, যার সঙ্গে দুর্বল অনুবাদস্পৃহা যুক্ত হওয়ায় পাঠক তার দ্বারা খুব কমই উপকৃত হ'ন।

আমাদের লক্ষ্য পূরণ করতে পারেন এমন একজনকে খুঁজতে গিয়ে পশ্চিমবঙ্গ, নেপাল, বাঙলাদেশ এবং বহির্বাঙলার বাঙলাভাষী ইলেকট্রনিক্স অনুরাগী ছাত্র, শিক্ষক ও সাধারণ আগ্রহীদের একান্ত বিশ্বাসভাজন ও অতি প্রিয় সৌম্য মিত্রকেই আমরা বেছে নিই। ভুল যে করিনি, এই গ্রন্থটিই তার সাক্ষাৎ প্রমাণ।

সৌম্য মিত্রের পরিচয় ইলেকট্রনিক্স অনুরাগীদের কাছে নতুন করে জানানোর কিছু নেই। সৌম্য মিত্র বাঙলাভাষায় ইলেকট্রনিক্স চর্চার পথিকৃত। তাঁর সম্পাদিত ইলেকট্রনিক্স ও ফোটোগ্রাফির একমাত্র সচিত্র মাসিক পত্রিকা 'ফোটন' আজ বাঙলায় ইলেকট্রনিক্স চর্চার ক্ষেত্রে রীতিমত আলোড়ন সৃষ্টি করেছে। তাঁর কাছ থেকে প্রেরণা পেয়ে একাধিক ইলেকট্রনিক্স চর্চার ক্লাব গড়ে উঠেছে ও উঠছে। নতুন পত্রিকার আবির্ভাব ঘটছে। বাঙলাভাষী শিক্ষার্থীরা এগিয়ে আসছেন ইলেকট্রনিক্স বিদ্যাকে আয়ত্ত করার সহজ পদ্ধতির সন্ধান পেয়ে। অনেকে রোজগারের নতুন পথ-নির্দেশ পেয়েছেন তাঁর রচনা পড়ে। তাঁর ভাষা, উপস্থাপনা-রীতি, শিক্ষাদানের অনন্য সাধারণ দক্ষতা এবং নিজে হাতেকলমে বিষয়টি আয়ত্ত করে তা শেখানোর পরিশ্রম স্বীকারের সততা, বস্তুতই আকর্ষণীয়। আমাদের লক্ষ্য পূরণে এজন্য তাঁকেই আমরা এই গ্রন্থের গ্রন্থকার, চিত্রকার, আলোকচিত্রী, পরিকল্পনাকারী, রূপসজ্জাশিল্পী, এক কথায় তাঁর ওপর বইটির প্রকাশনার ক্ষেত্রে সর্বৈবস্বাধীন দায়িত্ব অর্পণ ক'রে এবং তাঁর কাছ থেকে বস্তুত সার্থক এই অনবদ্য গ্রন্থটি পেয়ে তা শিক্ষার্থীদের হাতে তুলে দিতে পেরে যথার্থ খুশি হতে পেরেছি।

এই বইয়ে কম্পিউটারের প্রাযুক্তিক দিক বা হার্ডওয়্যার এবং ব্যবহারিক বিদ্যা বা সফটওয়্যার বিষয় দুটি তো শেখানো হয়েইছে, উপরন্তু লেখক বইটির শেষ পরিচ্ছেদে যোগ করেছেন মহামূল্য একটি পারিভাষিক অভিধান। এ বইয়ের বিষয়াবলী বইয়ের মুখবন্ধ অনুচ্ছেদে আলোচিত হয়েছে।

— রতন বিশ্বাস, প্রকাশক